THREE-DIMENSIONAL FINITE-ELEMENT ANALYSIS INCLUDING ELASTIC-PLASTIC MATERIAL RESPONSE

by

Fritz Hatt

Dissertation submitted to the Graduate Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Civil Engineering

APPROVED:

R. M. Barker, Chairman

D. Frederick

G. A. Gray

C. T. Herakovich

S. M. Holzer

July, 1974

Blacksburg, Virginia

ACKNOWLEDGMENTS

The author wishes to express his appreciation to Dr. Richard M. Barker, his major professor, for his assistance and guidance throughout the past several years.

The investigation presented herein was supported by the Department of Defense, Project THEMIS, Contract Number DAA F07-69-0444, with Watervliet Arsenal, Watervliet, New York. This financial assistance is gratefully acknowledged.

TABLE OF CONTENTS

TABLE OF CONTENTS - continued

LIST OF FIGURES

LIST OF FIGURES - continued

I. INTRODUCTION

Throughout the past decade, the finite-element analysis firmly established itself, particularly in the field of structural analysis. Membranes and frames were among the first structures to be examined with the aid of discrete elements. More involved element models were required for plates and shells. Three-dimensional elements were developed by the same principles governing two-dimensional models. Extensions into field problems such as heat conduction and seepage were made simultaneous to the intensive efforts carried out in the area of structural analysis. Another domain of interest was the dynamic analysis capability.

Almost from the beginning, the finite-element analysis was used to investigate nonlinear problems. For most practical problems exhibiting nonlinearity, the finite-element method proved itself as a versatile and powerful tool. Areas such as nonlinear material response, geometric and dynamic nonlinearities were usually treated separately. Plasticity, creep and visoelasticity were phenomenons that fell under the category of nonlinear material response. Geometrical nonlinearities took into account large deflections. Nonlinear dynamics solutions included large deflections and elastic-plastic materials under dynamic loads.

In terms of the solution strategy all analysis schemes can be separated into two parts: the linearization problem and the state determination problem. This investigation addresses itself to the problem of a nonlinear material response, where a solution can be found by an "initial stress" approach or by a "tangent modulus" concept. The first one, alternately called "residual force" method, places the emphasis on the state determination. For given strains the correct elastic-plastic stresses are evaluated. In the "tangent modulus" approach the computational effort is largely concentrated in the linearization. This usually means that a new force-displacement relationship reflecting the nonlinear characteristics has to be found repeatedly.

In this study special attention is paid to the initial stress approach. These stresses are equivalent to the so-called residual forces which are introduced to maintain equilibrium at all times. They are redistributed through an iterative scheme which assures that their magnitude decreases until they become negligible. One such iterative scheme is the Newton-Raphson method which amounts to an evaluation of the tangent stiffness matrix for every iterative step. If during the iterative process the original stiffness matrix is used exclusively, the term constant stiffness iteration applies. A modified Newton-Raphson approach where the total stiffness matrix is recalculated only at the beginning of a new load increment is also available.

The major advantage of the constant stiffness iteration is that the original stiffness matrix can be reused in its decomposed form which is available after the first linear solution. For problems in which the plastic region covers a large part of the total domain, the elastic stiffness matrix does not give a good representation of the nonlinear response, and therefore, convergence may become slow. For this reason, an acceleration scheme is proposed and implemented in the computational procedure.

With the aid of the computer program described in this investigation, problems that do not lend themselves to a plane strain, plane stress, or axisymmetric idealization, but require modelling in three dimensions, will be investigated. In addition to the areas of three-dimensional plasticity discussed herein, the computer analysis will be able to provide valuable information and added insight for a broad class of nonlinear material problems.

II. LITERATURE REVIEW

In most books and publications on the finite-element method, the paper by Turner et al. [1] applying plane triangular elements to a stiffness analysis in plane stress, is usually considered to be the groundstone to the finite-element analysis. The constant strain triangle was soon followed by new elements with more degrees-of-freedom often developed with respect to specific applications, such as plates and shells.

The first successful nonlinear investigations on simple problems were carried out in the field of plasticity. Ilyushin [2] established the method of successive elastic solutions. With the introduction of the digital computer this method became applicable to any problem whose elasticity solution was available. A. Mendelsson and S. S. Manson [3] solved a variety of elastic-plastic problems by the method of successive elastic solutions.

In the context of the finite-element analysis it was Argyris [4] and Gallagher et al. [5] who first implemented the "initial strain" approach. Zienkiewicz, Valliappan and King [6] showed that many problems dealing with nonlinear material response could be solved by the "initial stress" method. Pope [7], Swedlow and Young [8], and Marcal and King [9] developed the "tangent modulus" approach.

The element most widely used in these investigations was the constant strain triangle for its obvious simplicity in the application towards elastic-plastic behavior. The strains are monitored on an element-to-element basis and whenever the uniaxial or equivalent strain reaches the plastic region, the concerned elements will be governed by the elastic-plastic constitutive relationship which was explicitly given by Yamada et al. [10]

For applications towards beams and plates, Armen, Pifko and Levine [11] developed the capability for a beam or a rectangular plate element to allow a yield front to progress through the thickness of the element. Applications of elastic-plastic material response with the use of higher order, isoparametric elements have been presented by Nayak and Zienkiewicz [12]. A thermoelastoplastic and creep analysis using a three-dimensional 24-DOF isoparametric element was developed by Yagawa and Ando [13]. Sharifi and Yates [14] presented a computer program for thermoelastoplasticity with creep, capable of modelling a continuum by several types of elements.

The situation where a nonlinear solution is achieved by successive iterations, opens a whole array of possibilities as to how the iteration is to be performed and how to ensure reliability and speed of convergence. The convergence of an effective-force and material-stiffness iteration was discussed by Havner [15]. Acceleration of iterative matrix processes in general and with respect to elastic-plastic analysis were discussed by Jennings [17] and Boyle and Jennings [18].

As for examples and applications, problems such as the notched bar, perforated strip, deep beam, and problems in rock mechanics, and thermal stress have been solved successfully with the described methods. Micromechanics of fiber reinforced composites have also drawn a lot of attention. The problem of transverse normal loading was treated by Adams [20], and the longitudinal loading was focused on by Lin et al. [21]. Post-yielding behavior in composites was discussed by Foye [22,23]. A two-dimensional investigation into bond failures with an elastic-plastic bond element was carried out by Owen [24]. Three-dimensional experimental results for temperature stresses were reported by Asamoah and Wood [25,26].

The area of three-dimensional nonlinear finite-element analysis is addressed in two of the publications [13,14] discussed above. Both are characterized by the step-by-step analysis required for the modelling of time dependent phenomena, such as temperature flow and creep. The iterative procedures within one step do not make use of any acceleration. Sharifi and Yates [14] included an unspecified continuum element in their program, and Yagawa and Ando [13] used a 24-DOF isoparametric hexahedron.

The nonlinear finite-element program developed in this study based on the constant stiffness method expands the development of Nayak and Zienkiewicz [12] to three dimensions. The element used here is the 60-DOF isoparametric hexahedron. Applications, however, are restricted to elastic-plastic material behavior. A new acceleration scheme is also discussed and incorporated in the computer program.

III. THEORY OF THE FINITE-ELEMENT METHOD

The formulation of a finite-element problem can be broken up into three distinct steps. They are: (a) idealization, (b) element model, and (c) assembly of the total system. This procedure yields a system of linear simultaneous equations. With the aid of one of several available solution methods, the primary unknowns can be determined. Most frequently used is the modified Gauss elimination scheme taking advantage of the symmetry and the banded form of the coefficient matrix.

A. Idealization

In structural applications, the idealization process can be viewed as the division of a continuum into a number of finite-elements. This rather important aspect of the problem formulation, however, is not investigated in this study. Intuitively it is clear that in locations of special interest within the continuum the number of elements should be increased. Where large stress gradients are anticipated, a larger number of elements seems appropriate. This leads to a better reproduction of the corresponding strain variations. In applications where an element model capable of representing only a state of constant strain over the element is used, the density of elements should reflect the magnitude of the changes of the strains. Also, it is of some advantage that in areas which demand special attention an increased number of elements furnishes additional stresses and strains. Sometimes a larger number of elements becomes mandatory just to obtain a satisfactory geometrical representation of the continuum.

B. Element Model

In modelling the element response, the displacement formulation is almost exclusively used. This means that the nodal displacements are chosen as the extrinsic variables. They are related to the nodal forces through the element stiffness matrix. By minimizing the total potential energy, or by using a direct approach as outlined below, the expression for the element stiffness matrix can be obtained.

With the selection of an element type and the associated degrees-of-freedom, an appropriate interpolation function, relating the extrinsic variables at the nodal points to those at a general location, is of the form

$$\{f\} = [N]\{\delta\}^e \tag{3.1}$$

If displacements are chosen as degrees-of-freedom of the element, $\{\delta\}^e$ represents the vector of the nodal displacements, $\{f\}$ the displacements at a general point, and $[N]$ the interpolation function. In this context, $[N]$ is also called, more specifically, the displacement function.

The strains at any point of the element are now found by proper differentiation of the displacements. In a general three-dimensional situation with only the linear strain terms included, the following expression holds.

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \varepsilon_{xy} \\ \varepsilon_{yz} \\ \varepsilon_{zx} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial w}{\partial z} \\ \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \\ \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \\ \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \end{Bmatrix} \tag{3.2}$$

With Eq. 3.1 substituted for u, v, and w, the strain-displacement relationship can be written.

$$\{\varepsilon\} = [B]\{\delta\}^e \tag{3.3}$$

The constitutive relationship incorporating Hooke's law is denoted in matrix form as follows.

$$\{\sigma\} = [D]\{\varepsilon\} \tag{3.4}$$

where [D] is the elasticity matrix that is determined by the elastic properties of the material.

By imposing a virtual displacement $\{\delta^*\}^e$ on the element, the internal work U done by the stresses has to equal the external work W performed by the loads, if equilibrium is to be satisfied.

$$U = \int_V \{\varepsilon^*\}^T\{\sigma\}dV = \int_V (\{\delta^*\}^e)^T\, [B]^T\{\sigma\}^e\, dV$$

$$= \int_V (\{\delta^*\}^e)^T\, [B]^T[D]\{\varepsilon\}^e dV = \int_V (\{\delta^*\}^e)^T[B]^T[D][B]\{\delta\}^e dV \tag{3.5}$$

$$W = (\{\delta^*\}^e)^T\{F\}^e \tag{3.6}$$

After setting U equal to W, the element stiffness matrix $[k]^e$ can be determined as follows.

$$\{F\}^e = \int_V [B]^T[D][B]\{\delta\}^e dV = [k]^e\{\delta\}^e \qquad (3.7)$$

$$[k]^e = \int_V [B]^T[D][B]dV \qquad (3.8)$$

C. Assembly and Solution

By enforcing the equilibrium condition that the sum of the nodal forces is equal to the sum of the external forces at all nodal points, the system of equations representing the total stiffness matrix can be assembled.

$$\{R_i\} = \sum \{F_i\} \qquad (3.9)$$

Eq. 3.9 contains the same number of linear equations as there are degrees-of-freedom for the nodal point i. After enforcing the prescribed boundary conditions on the appropriate displacements, the system of equations becomes non-singular and can therefore be solved for the unknowns.

After the evaluation of the primary unknowns, the displacements, from

$$\{R\} = [K]\{\delta\} \qquad (3.10)$$

the secondary unknowns, the stresses, are found on an element-by-element basis.

$$\{\sigma\}^e = [D]\{\varepsilon\}^e = [D][B]\{\delta\}^e \qquad (3.11)$$

This can be verified by combining Eqs. 3.3 and 3.4.

D. Equivalent Nodal Forces

Since the total system equations are found by enforcing the balance condition at the nodal points, it is obvious that all external loadings will have to be expressed in the form of nodal forces. General load conditions such as body forces, boundary pressures and initial strains, therefore, must be converted to equivalent nodal forces.

By imposing a virtual displacement $\{\delta^*\}^e$, the external work done by the equivalent nodal forces has to be equal to the negative internal work performed by the stresses. If it is assumed that $\{\bar{p}\}$ represents the body load at a general point of the element and that $\{\varepsilon_0\}$ stands for a state of initial strain, the external work expression W can be written.

$$W = (\{\delta^*\}^e)^T\{F\}^e_p + (\{\delta^*\}^e)^T\{F\}^e_{\varepsilon_0} \tag{3.12}$$

The displacements and the strains due to $\{\delta^*\}^e$ can be obtained from Eq. 3.1 and Eq. 3.4.

$$\{f^*\} = [N]\{\delta^*\}^e \tag{3.13}$$

$$\{\varepsilon^*\} = [B]\{\delta^*\}^e \tag{3.14}$$

The internal work U due to $\{\bar{p}\}$ and $\{\varepsilon_0\}$ can be expressed as follows.

$$U = \int_V (\{f^*\}^T\{\bar{p}\} + \{\varepsilon^*\}\{\sigma_0\})\, dV$$

$$= \int_V [N]\{\delta^*\}^e\{\bar{p}\}dV + \int_V [B]\{\delta^*\}^e\{\sigma_0\}dV$$

Considering that $\{\sigma_0\} = [D]\{\varepsilon_0\}$ and by factoring out $\{\delta^*\}^e$ we obtain

$$U = (\{\delta^*\}^e)^T \left(\int_V [N]^T\{\bar{p}\}dV + \int_V [B]^T[D]\{\varepsilon_0\}dV\right) \tag{3.15}$$

Since the virtual displacement $\{\delta^*\}^e$ represents an arbitrary multiplier, the following can be stated after setting W equal to -U.

$$\{F\}^e_p = -\int_V [N]^T\{\bar{p}\}\ dV \tag{3.16}$$

$$\{F\}^e_{\varepsilon_0} = -\int_V [B]^T[D]\{\varepsilon_0\}\ dV \tag{3.17}$$

In the case of initial stress Eq. 3.17 simply becomes

$$\{F\}^e_{\sigma_0} = -\int_V [B]^T\{\sigma_0\}\ dV \tag{3.18}$$

Equivalent nodal forces for boundary pressure $\{\bar{g}\}$ are found as a special case of Eq. 3.16, with the integration carried out only over the loaded surface.

$$\{F\}^e_g = -\int_S [N]^T\{\bar{g}\}\ dS \tag{3.19}$$

An additional observation has to be made with respect to the calculation of the stresses $\{\sigma\}$. The initial strain state $\{\varepsilon_0\}$, replaced by the equivalent forces $\{F\}^e_{\varepsilon_0}$, has to be subtracted from the strains obtained as a result of $\{F\}^e_{\varepsilon_0}$.

$$\{\sigma\} = [D](\{\varepsilon\}-\{\varepsilon_0\}) \tag{3.20}$$

For the case of an unrestrained continuum it is obvious that an imposed state of strain $\{\varepsilon_0\}$ should not produce any stresses.

IV. THE 20-NODE HEXAHEDRON

In this chapter the necessary steps for the evaluation of the element stiffness matrix of a curvilinear 60-DOF hexahedron are described. The calculation of equivalent nodal forces due to either body loads or surface pressure is also presented. In a last step the element stresses are expressed as a function of the known nodal displacements.

A. The Coordinate System

By establishing a natural coordinate system (ξ,η,ζ) as shown in Fig. 4.1 with the origin at the center point of the general element, it is possible to relate the element under consideration to its "parent" element by means of a mapping process. The arbitrarily shaped, often referred to as the "derived" element, in the (x,y,z) system is projected into a cube in the (ξ,η,ζ) system with its faces represented by the planes $\xi = \pm 1$, $\eta = \pm 1$, and $\zeta = \pm 1$. The mapping functions relating the two coordinate systems are expressed as follows:

$$\begin{aligned} x &= f(\xi,\eta,\zeta) = N_1x_1 + N_2x_2 + \ldots = \{N\}^T\{x_n\} \\ y &= g(\xi,\eta,\zeta) = N_1y_1 + N_2y_2 + \ldots = \{N\}^T\{y_n\} \\ z &= h(\xi,\eta,\zeta) = N_1z_1 + N_2z_2 + \ldots = \{N\}^T\{z_n\} \end{aligned} \tag{4.1}$$

$\{x_n\}$, $\{y_n\}$, and $\{z_n\}$ represent the nodal coordinates, and $\{N\}^T$ is dependent on ξ, η, and ζ. Since the shape functions relating the displacements within the element to the nodal displacements will have the same parametric representation, the element is called isoparametric. The evaluation of the N_i's will be carried out in the following section.

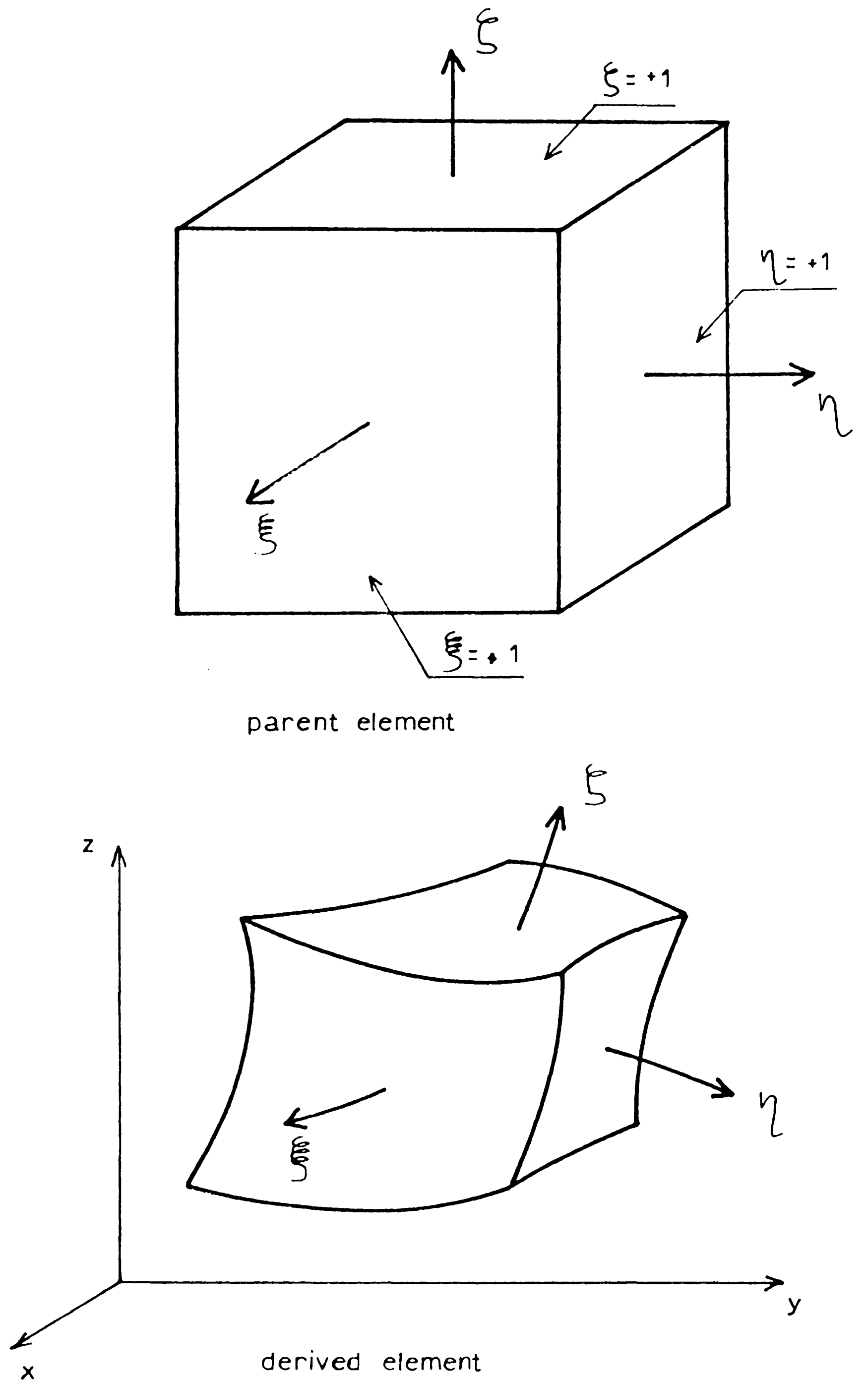


FIG. 4.1. The parent and the derived element in the natural and the cartesian coordinate system

The most important consideration here is that the mapping establishes a one-to-one relationship between the derived and the parent element. This means that the mapping will be successful except in cases where the derived element possesses regions of overlap or undue distortion. The necessary and sufficient condition for a one-to-one relationship is that the Jacobian determinant

$$J = \frac{\partial(x,y,z)}{\partial(\xi,\eta,\zeta)} = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{vmatrix} \tag{4.2}$$

does not vanish anywhere within the domain of the element.

B. The Shape Function

The relationship between displacements at an arbitrary location (ξ,η,ζ) within the element and the nodal displacements is expressed in the most general form with the aid of a shape or interpolation functions as follows:

$$\begin{aligned} u(\xi,\eta,\zeta) &= N_1u_1 + N_2u_2 + \ldots = \{N\}^T\{u_n\} \\ v(\xi,\eta,\zeta) &= N_1v_1 + N_2v_2 + \ldots = \{N\}^T\{v_n\} \\ w(\xi,\eta,\zeta) &= N_1w_1 + N_2w_2 + \ldots = \{N\}^T\{w_n\} \end{aligned} \tag{4.3}$$

In a first step we express the u-displacement in polynomial form

$$u = \alpha_1 + \alpha_2\xi + \ldots = [1,\xi,\eta,\zeta\ldots]\ \{\alpha_n\} \tag{4.4}$$

Now the (ξ,η,ζ) values of all twenty nodal points can be substituted into Eq. 4.4 and a set of equations of the following form is obtained.

$$\{u_n\} = [C]\{\alpha_n\} \tag{4.5}$$

With the [C] matrix known, the α's can now be written as

$$\{\alpha_n\} = [C]^{-1}\{u_n\} \tag{4.6}$$

Substituted back into Eq. 4.4, u can be expressed as

$$u = [1,\xi,\eta,\zeta,\ \ldots][C]^{-1}\{u_n\} = \{N\}^T\{u_n\} \tag{4.7}$$

From Eq. 4.7 it becomes obvious that

$$\{N\}^T = [1,\xi,\eta,\zeta\ldots][C]^{-1} \tag{4.8}$$

The above derivation implies that the polynomial expansion will have 20 terms, as many as there are nodal points. Obviously, the choice of terms to be included in the expansion becomes quite important. The overriding consideration is the following one: The variation along one of the coordinate directions ξ, η, or ζ with the other two held constant, should be only of second order. This restriction ensures that along all element edges, determined by two corner nodes and one midside node, only one curve representing the displaced element edge is possible. This is important in the case of adjoining elements with common edges. Inter-element gaps or overlaps are therefore avoided if common edges remain common in the displaced state.

A complete polynomial in three variables can be represented by a tetrahedron with components of different order located on different planes.

1st order

$$\begin{array}{ccc} & x & \\ y & & z \end{array}$$

2nd order

$$\begin{array}{ccccc} & & x^2 & & \\ & xy & & xz & \\ y^2 & & yz & & z^2 \end{array}$$

3rd order

$$\begin{array}{ccccccc} & & & x^3 & & & \\ & & x^2y & & x^2z & & \\ & xy^2 & & xyz & & xz^2 & \\ y^3 & & y^2z & & yz^2 & & z^3 \end{array}$$

4th order

$$\begin{array}{ccccccccc} & & & & x^4 & & & & \\ & & & x^3y & & x^3z & & & \\ & & x^2y^2 & & x^2yz & & x^2z^2 & & \\ & y^3x & & y^2xz & & z^2xy & & xz^3 & \\ y^4 & & y^3z & & y^2z^2 & & yz^3 & & z^4 \end{array}$$

The complete 4th order polynomial has a total of 35 terms. Twelve of those are deleted because they violate the condition of quadratic variation only in one of the variables with the other two constant. By leaving out the x^2y^2, y^2z^2 and z^2x^2 terms we have now left an expansion with the desired number of terms.

This incomplete 4th order polynomial yields a non-singular [C] matrix which is, of course, essential to the derivation of the shape functions. (For convenience, the terms of $[C]^{-1}$ are given in Appendix 1.)

One other important aspect in the derivation of the shape functions is to ensure that for rigid body displacements no straining occurs. This constant strain condition is satisfied for the given shape function. A general shape function producing constant strain can be written as

$$u = a + bx + cy + dz$$

If this form is substituted into the one previously established as

$$u = \{N\}^T\{u_n\} = N_1u_1 + N_2u_2 + \dots$$

the following can be written:

$$\begin{aligned} &N_1(a+bx_1+cy_1+dz_1) + N_2(a+bx_2+cy_2+dz_2) + \dots \\ &\quad = a(N_1+N_2+\dots) + b(N_1x_1+N_2x_2+\dots) \\ &\qquad + c(N_1y_1+N_2y_2+\dots) + d(N_1z_1+N_2z_2+\dots) \\ &\quad = a + bx + cy + dz \end{aligned}$$

For this to be true, the sum of the N's has to be equal to one. This can be checked easily by summing up the lines of the $[C]^{-1}$ matrix. The first line adds up to one and the sums of the remaining lines are zero.

C. The Strain-Displacement Relationship

As pointed out in Chapter III, the strains are obtained through proper differentiation of the displacements.

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \varepsilon_{xy} \\ \varepsilon_{yz} \\ \varepsilon_{zx} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial w}{\partial z} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\ \frac{\partial w}{\partial x} + \frac{\partial y}{\partial z} \end{Bmatrix} \tag{4.9}$$

The general displacements u, v, and w are expressed in terms of the nodal displacements by means of the shape function and the differentiation is carried out to give the strains

$$\{\varepsilon\} = [B_1, B_2, \ldots]\{\delta\}^e = [B]\{\delta\}^e \tag{4.10}$$

The submatrix $[B_i]$ takes on the following form:

$$[B_i] = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_i}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_i}{\partial z} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} & 0 & \frac{\partial N_i}{\partial x} \end{bmatrix} \tag{4.11}$$

The problem that poses itself at this stage is that the partials of the N_i's with respect to x, y, and z are not directly available, since the N_i's are the functions of ξ, η, and ζ.

In this case the mathematical rules for "composite functions" are applicable [27]. N_i can be written in the following fashion.

$$N_i = F(x,y,z) = G(\xi,\eta,\zeta) = F[f(\xi),g(\eta),h(\zeta)] \tag{4.12}$$

The partial derivatives of G with respect to ξ, η, and ζ can be found by means of the chain-rule.

$$\begin{aligned} \frac{\partial G}{\partial \xi} &= \frac{\partial F}{\partial x}\frac{\partial f}{\partial \xi} + \frac{\partial F}{\partial y}\frac{\partial g}{\partial \xi} + \frac{\partial F}{\partial z}\frac{\partial h}{\partial \xi} \\ \frac{\partial G}{\partial \eta} &= \frac{\partial F}{\partial x}\frac{\partial f}{\partial \eta} + \frac{\partial F}{\partial y}\frac{\partial g}{\partial \eta} + \frac{\partial F}{\partial z}\frac{\partial h}{\partial \eta} \\ \frac{\partial G}{\partial \zeta} &= \frac{\partial F}{\partial x}\frac{\partial f}{\partial \zeta} + \frac{\partial F}{\partial y}\frac{\partial g}{\partial \zeta} + \frac{\partial F}{\partial z}\frac{\partial h}{\partial \zeta} \end{aligned} \tag{4.13}$$

The above Eqs. 4.13 can be expressed in matrix form as

$$\begin{Bmatrix} \frac{\partial G}{\partial \xi} \\ \frac{\partial G}{\partial \eta} \\ \frac{\partial G}{\partial \zeta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial \xi} & \frac{\partial g}{\partial \xi} & \frac{\partial h}{\partial \xi} \\ \frac{\partial f}{\partial \eta} & \frac{\partial g}{\partial \eta} & \frac{\partial h}{\partial \eta} \\ \frac{\partial f}{\partial \zeta} & \frac{\partial g}{\partial \zeta} & \frac{\partial h}{\partial \zeta} \end{bmatrix} \begin{Bmatrix} \frac{\partial F}{\partial x} \\ \frac{\partial F}{\partial y} \\ \frac{\partial F}{\partial z} \end{Bmatrix} = [J^T] \begin{Bmatrix} \frac{\partial F}{\partial x} \\ \frac{\partial F}{\partial y} \\ \frac{\partial F}{\partial z} \end{Bmatrix} \tag{4.14}$$

For simplicity $[J^T]$, the transpose of the Jacobian matrix, is introduced. Usually, the Jacobian is encountered in the form of a determinant (Eq. 4.2) as given in Section A of this chapter. The partial derivatives with respect to x, y, and z can now be found from Eq. 4.14.

$$\begin{Bmatrix} \frac{\partial F}{\partial x} \\ \frac{\partial F}{\partial y} \\ \frac{\partial F}{\partial z} \end{Bmatrix} = [J^T]^{-1} \begin{Bmatrix} \frac{\partial G}{\partial \xi} \\ \frac{\partial G}{\partial \eta} \\ \frac{\partial G}{\partial \zeta} \end{Bmatrix} \tag{4.15}$$

and after substituting N_i for the generalized G and F functions, the desired partials can be written

$$\begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} = [J^T]^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{Bmatrix} \qquad (4.16)$$

The individual terms of the Jacobian matrix in Eq. 4.14 are of the form

$$\frac{\partial f}{\partial \xi} = \frac{\partial x}{\partial \xi} \qquad , \text{ where } x = \{N\}^T\{x_n\}$$

$$\frac{\partial x}{\partial \xi} = \{\frac{\partial N^T}{\partial \xi}\} \{x_n\}$$

With all the $\frac{\partial N_i}{\partial \xi}$, $\frac{\partial N_i}{\partial \eta}$, and $\frac{\partial N_i}{\partial \zeta}$ known, the transpose of the Jacobian matrix can be inverted and the partial derivatives of the shape functions with respect to the global coordinates (x,y,z) can be found as indicated in Eq. 4.16. For more detail see Appendix 2, where the expressions for the partial derivatives of {N} with respect to ξ, η, and ζ, and the terms of the Jacobian matrix are listed.

D. Evaluation of the Element Stiffness Matrix

From Chapter III, Eq. 3.7, the element stiffness matrix is given by

$$\int_V [B^T][D][B] \, dV$$

The above volume integral can be carried out numerically by Gaussian-type integration rules or by Gaussian quadrature. For the element used in this study the choice was narrowed down to the 14-point rule given originally by Hammer and Stroud [28] and the 3x3x3 Gauss scheme. Irons [29] stated that the 14-point rule was accurate to the complete quintic and that the sextic errors were moderately small compared to the 27-point Gauss rule. A comparison of the two rules was made by T. K. Hellen [30]. For a cantilever problem the 14-point rule showed small percentage errors compared to the Gauss quadrature giving exact results. For a moderately thin cylinder, the two rules gave almost identical percentage errors and in the case of extremely thin cylinders under internal pressure the 14-point rule gave superior results. Since the overall computational effort for all steps involving numerical integration is only about 50 percent for the 14-point rule as compared to the 27-point Gauss rule, the moderately small errors in accuracy were outweighed by the gain in efficiency and therefore the 14-point rule was adopted. The location of the individual integration points with respect to the parent element are shown in Fig. 4.2 and the mathematical formulation including integration point coordinates and weight factors is given below

$$\begin{aligned}\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta,\zeta)\,d\xi\, d\eta\, d\zeta = {} & B_6\{f(-b,0,0)+f(b,0,0)+f(0,-b,0) \\ & +f(0,b,0)+f(0,0,-b)+f(0,0,b)\} \\ & + C_8\{f(-c,-c,-c)+f(c,-c,-c)+f(c,c,-c) \\ & +f(c,c,c)+f(-c,c,-c)+f(-c,-c,c) \\ & +f(-c,c,c)+f(c,-c,c)\} \qquad (4.17)\end{aligned}$$

where $B_6 = 0.886426593$, $b = 0.795822426$

$C_8 = 0.335180055$, $c = 0.758786911$

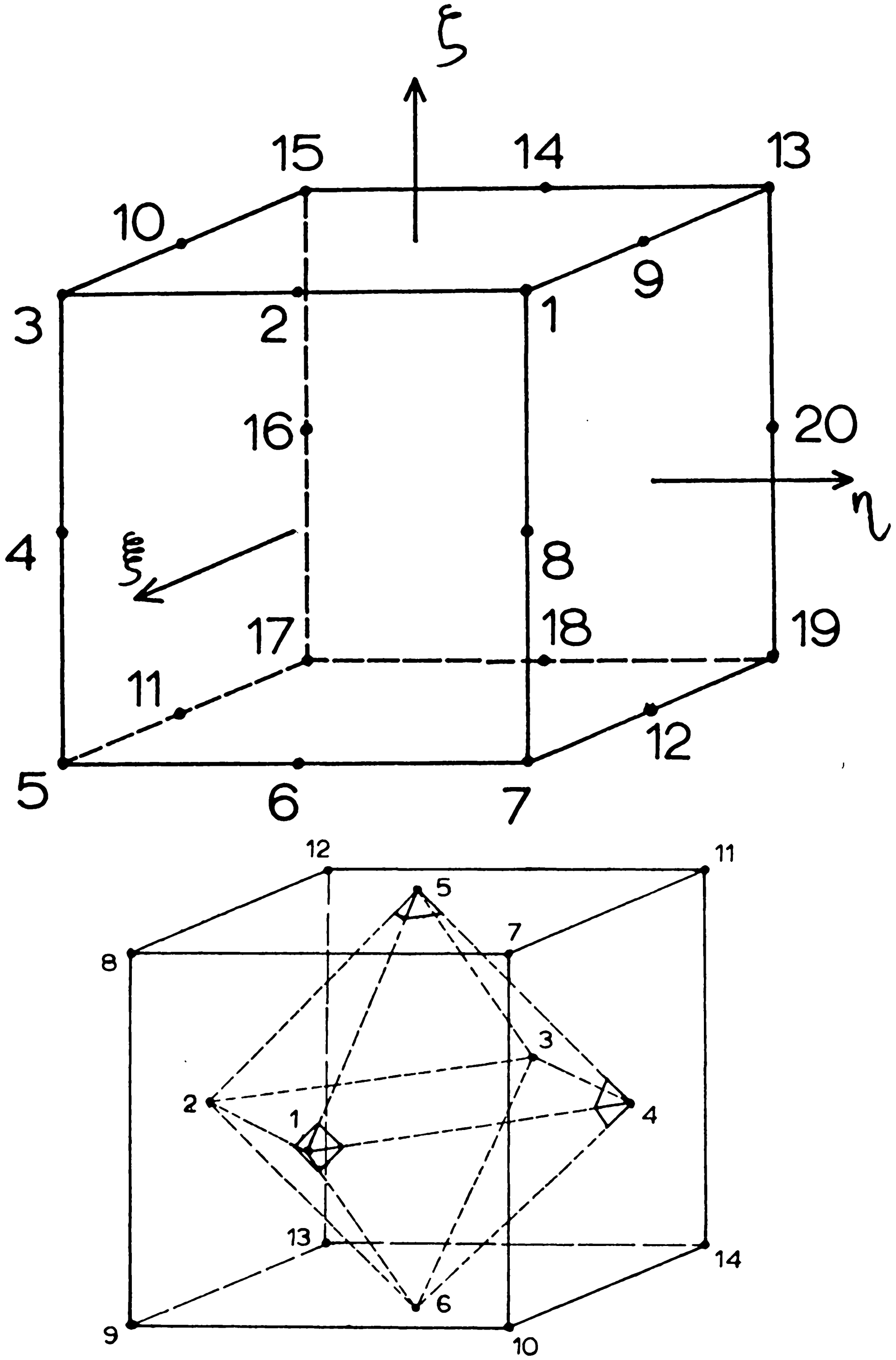


FIG.4.2. Locations and numbering of nodal points and integration points

As the above equation indicates, the integration is carried out over the limits of the parent element in the (ξ,η,ζ) system. Under the condition that the mapping (see Section A of this chapter) be one-to-one, i.e., the determinant of the Jacobian never vanishes throughout the region of the element, the following relationship governing the volumetric integration holds:

$$\iiint_V F(x,y,z)dxdydz = \iiint_{V'} F(x,y,z) \left|\frac{\partial(x,y,z)}{\partial(\xi,\eta,\zeta)}\right| d\xi d\eta d\zeta \qquad (4.18)$$

In the evaluation of the element stiffness matrix the function $F(x,y,z)$ represents the matrix triple product $[B^T][D][B]$. This matrix product evaluated at the integration points and subsequently multiplied by the appropriate determinant of the Jacobian then represents the f-contributions in Eq. 4.17.

E. Nodal Forces Due to Initial Stress, Initial Strain

The equivalent nodal forces, representing states of initial stress or initial strain, can be calculated as indicated in Chapter III.

$$\{F\}^e_{\varepsilon_o} = -\int_V [B]^T[D]\{\varepsilon_o\}\, dV \qquad (3.17)\text{ rep.}$$

$$\{F\}^e_{\sigma_o} = -\int_V [B]^T\{\sigma_o\}\, dV \qquad (3.18)\text{ rep.}$$

For those volumetric integrations the same rules as outlined for the evaluation of the element stiffness matrix apply. The numerical integration scheme remains the same and the same mapping functions still govern the transformation from the (x,y,z) system to the (ξ,η,ζ)

coordinates. The limits of integration are given by -1 and +1 in the ξ, η, and ζ directions. The function to be integrated, again, has to be multiplied by the appropriate determinant of the Jacobian.

Nodal forces due to a change of temperature can be found as a special case of the initial strain forces. With the coefficient of thermal expansion α and the change in temperature ΔT the initial strain vector $\{\varepsilon_o\}$ can be written as follows.

$$\{\varepsilon_o\} = \begin{Bmatrix} \alpha\ \Delta T \\ \alpha\ \Delta T \\ \alpha\ \Delta T \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

F. Nodal Forces Due to Body Loads

The body loads within one element can be expressed with the aid of the established shape functions as follows:

$$\{\bar{p}_x\} = \{N\}^T\{p_{x,n}\} \tag{4.19}$$

where $\{p_{x,n}\}$ is the vector containing the values of the body load at the nodal points for a given direction. Similar expressions hold for the components of $\{\bar{p}\}$ in y and z directions, the displacement functions remaining the same. The general expression (Eq. 3.16) given before can now be written as

$$\{F_x\}^e{}_p = -\int_V \{N\}^T\{\bar{p}_x\}\ dV \tag{4.20}$$

This integral is to be evaluated numerically in the same fashion as was done for the element stiffness matrix. For the force components in y and z directions the appropriate $\{\bar{p}\}$ has to be established from the nodal values as stated in Eq. 4.19.

G. Nodal Forces Due to Boundary Pressure

In many applications the external load may be represented by a surface pressure. If the element loaded in such a fashion is of irregular shape, an intuitive allocation of concentrated forces at the nodal points is no longer possible. The expression previously used for body loads can be modified as follows:

$$\{F_x\}^e{}_b = -\int_S \{N\}^T\{\bar{g}_x\}\, dS \tag{4.21}$$

where $\{\bar{g}_x\}$ represents the x-component of the surface pressure at a general point, and is related to the nodal values by means of the displacement function.

$$\{\bar{g}_x\} = \{N\}^T\{g_n\} \tag{4.22}$$

As is obvious from Eq. 4.21 the integration is now carried out over a surface rather than the whole volume of the element.

In this derivation only the surface represented by $\xi = +1$ is considered to be loaded by a given pressure distribution. This surface, which is generally curved in the (x,y,z) system, will now be mapped into a square in the (η,ζ) plane. Expressed in parametric form, the following relationships can be written:

$$x = f(\eta,\zeta), \quad y = g(\eta,\zeta), \quad z = h(\eta,\zeta) \tag{4.23}$$

The surface area integrated over η, and ζ is given by the following [27].

$$S = \int_{-1}^{+1}\int_{-1}^{+1} \sqrt{EG-F^2}\ d\eta d\zeta \tag{4.24}$$

where

$$E = \left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2 + \left(\frac{\partial z}{\partial \eta}\right)^2$$

$$F = \frac{\partial x}{\partial \eta}\frac{\partial x}{\partial \zeta} + \frac{\partial y}{\partial \eta}\frac{\partial y}{\partial \zeta} + \frac{\partial z}{\partial \eta}\frac{\partial z}{\partial \zeta}$$

$$G = \left(\frac{\partial x}{\partial \zeta}\right)^2 + \left(\frac{\partial y}{\partial \zeta}\right)^2 + \left(\frac{\partial z}{\partial \zeta}\right)^2$$

All the partial derivatives contained in the above expressions are components of the Jacobian $\left|\frac{\partial(x,y,z)}{\partial(\xi,\eta,\zeta)}\right|$. Based on Eq. 4.24, the equivalent nodal forces due to boundary pressure can be written as

$$\{F_x\}^e_b = \int_{-1}^{+1}\int_{-1}^{+1} \{N\}^T\{\bar{g}_x\}\ \sqrt{EG-F^2}\ d\eta d\zeta \tag{4.25}$$

In carrying out the above integration the 3x3 Gauss quadrature formula is employed.

H. The Stress Evaluation

After the system of linear equations resulting from the assembly of the individual element stiffness matrices into the total stiffness matrix is solved for the primary unknowns, the displacements, the stresses are evaluated. The previously mentioned stress-strain relationship (Eq. 3.20) will now be reexamined.

$$\{\sigma\} = [D]\{\varepsilon-\varepsilon_0\} \qquad (3.20)\ \text{rep.}$$

Upon substitution of the strain-displacement equations (Eq. 4.10) it can be written as follows:

$$\{\sigma\} = [D][B]\{\delta\}^e - [D]\{\varepsilon_0\} \qquad (4.26)$$

In this study, the element stresses are evaluated at the 14 integration points rather than at the nodal points. The reason for this is to maintain consistency in the analysis. This will become obvious during the discussion of the solution of the inelastic problem.

The stress evaluation becomes straightforward when both the constitutive matrix [D], and the strain-displacement matrix [B] are known. The [B] matrices evaluated at the individual integration points, previously used during the calculation of the element stiffness matrix, are regenerated for the evaluation of the stresses.

V. THE ELASTIC-PLASTIC RESPONSE

Incorporated in the theory of plasticity, describing the material behavior beyond the elastic range, are two important concepts. The first one is the yield condition. Usually given as a function of the invariants of the stress deviator tensor, the yield condition determines whether a given stress state still falls within the elastic regime, or whether plastic flow is taking place. The second concept, the flow rule, supplements the elastic constitutive relationship based on Hooke's law. It relates the plastic strain increment to the stress increment during the event of plastic flow. This investigation is restricted to isotropic strain hardening behavior.

A. The Yield Criterion

In a uniaxial stress situation which can be modelled by the simple tensile test it is possible to find the yield stress σ_Y at which the linear elastic response is replaced by the elastic-plastic response. In a general three-dimensional stress state, however, the yield condition determines what combination of the six stresses separates the elastic from the elastic-plastic behavior. For an isotropic material the invariants of the stress tensor are used more conveniently. Experimental investigations have shown that the hydrostatic or spherical state of stress has no appreciable influence on plastic flow. This suggests that a general yield criterion can be expressed as a function of the invariants of the stress deviator tensor listed below.

$$I_1' = \sigma_x' + \sigma_y' + \sigma_z' = 0 \tag{5.1}$$

$$I_2' = \frac{1}{2}\sigma_{ij}'\sigma_{ij}' = \frac{1}{2}(\sigma_x'^2+\sigma_y'^2+\sigma_z'^2)$$

$$+ \sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2$$

$$= \frac{1}{6}[(\sigma_x-\sigma_y)^2+(\sigma_y-\sigma_z)^2+(\sigma_z-\sigma_x)^2] + \sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2 \tag{5.2}$$

$$I_3' = \frac{1}{3}\sigma_{ij}\sigma_{jk}\sigma_{ki} \tag{5.3}$$

By using the invariants of the stress deviator tensor rather than those of the stress tensor, the spherical state of stress is excluded from the yield condition because I_1' is identically zero. Hence, a general yield criterion for an isotropic material can be written as

$$f(I_2', I_3') = 0 \tag{5.4}$$

In order to simplify things further, the dependence of f on I_3' can be neglected. This leads to a yield criterion solely dependent on I_2', a formulation that is well known as the von Mises condition.

$$f(I_2') = 0 \tag{5.5}$$

In a uniaxial stress state at yield described by

$$\sigma_x = \sigma_Y; \quad \sigma_y = \sigma_z = \sigma_{xy} = \sigma_{yz} = \sigma_{zx} = 0$$

or

$$\sigma_x' = \frac{2}{3}\sigma_Y; \quad \sigma_y' = \sigma_z' = -\frac{1}{3}\sigma_Y; \quad \sigma_{xy} = \sigma_{yz} = \sigma_{zx} = 0$$

the second invariant I_2' becomes from Eq. 5.2

$$I_2' = \frac{1}{3}\sigma_Y^2 = \frac{1}{3}\sigma_e^2 \tag{5.6}$$

The quantity σ_e is called the equivalent or effective stress which relates the general state of stress to the uniaxial state. The von Mises yield condition can now be written as

$$f(I_2') = \sigma_e^2 - \sigma_Y^2 = 0 \tag{5.7}$$

or

$$[\tfrac{1}{2}(\sigma_x-\sigma_y)^2+\tfrac{1}{2}(\sigma_y-\sigma_z)^2+\tfrac{1}{2}(\sigma_z-\sigma_x)^2] + 3\sigma_{xy}^2 + 3\sigma_{yz}^2 + 3\sigma_{zx}^2 - \sigma_Y^2$$
$$= 0 \tag{5.8}$$

Similarly, it can be stated that the distortion energy which is a function of the second invariant of the stress deviator tensor of the general stress state is equal to the distortion energy for the uniaxial stress state. This leads to the same definition of σ_e. The expression for the equivalent stress from Eq. 5.8 can also be written conveniently in tensor notation as follows.

$$\sigma_e = (\tfrac{3}{2}\,\sigma_{ij}'\sigma_{ij}')^{\frac{1}{2}} \tag{5.9}$$

B. The Elastic-Plastic Constitutive Relationship

For the development of a suitable flow rule, also accounting for hardening, a more general yield condition that includes effects of the total stresses σ_{ij}, total plastic strains ε_{ij}^p, and a hardening parameter k has to be considered.

$$g(\sigma_{ij},\varepsilon_{ij}^p,k) = 0 \tag{5.10}$$

The parameter k allows that the function g be dependent not only on the present state σ_{ij} and ε_{ij}^p, but also reflects a hardening history according to previous states of stress and strain. Suppose that Eq. 5.10 is satisfied, the total differential of g can be written as

$$dg = \frac{\partial g}{\partial \sigma_{ij}} d\sigma_{ij} + \frac{\partial g}{\partial \varepsilon_{ij}^{p}} d\varepsilon_{ij}^{p} + \frac{\partial g}{\partial k} dk \tag{5.11}$$

The yield condition (Eq. 5.10) with ε_{ij}^{p} and k held constant can be interpreted as a hypersurface in the multidimensional stress space. A change $dg < 0$ indicates a purely elastic change towards the inside of the yield surface. In the absence of plastic flow the increments of plastic strain $d\varepsilon_{ij}^{p}$ and the change of the hardening parameter dk are automatically zero. Therefore, in the case of unloading Eq. 5.11 reduces to

$$dg = \frac{\partial g}{\partial \sigma_{ij}} d\sigma_{ij} < 0 \tag{5.12}$$

The case of $dg = 0$ can be interpreted as neutral loading where no plastic strain changes occur and the hardening factor remains unchanged. For this case the following holds

$$dg = \frac{\partial g}{\partial \sigma_{ij}} d\sigma_{ij} = 0$$

Excluding the trivial case of $d\sigma_{ij} = 0$, the quantity $d\sigma_{ij}$ is tangent to the surface for neutral loading. For the vector product to be zero, $\frac{\partial g}{\partial \sigma_{ij}}$ has to be normal to the surface.

If $d\sigma_{ij}$ is pointing to the outside of the surface the vector product will be positive.

$$dg = \frac{\partial g}{\partial \sigma_{ij}} d\sigma_{ij} > 0 \tag{5.13}$$

This constitutes loading with plastic flow taking place. As already pointed out, this geometrical interpretation is made possible by assuming that for a stress increment $d\sigma_{ij}$ the plastic strains $\varepsilon_{ij}{}^p$ and the hardening parameter k remain constant. The flow rule connecting σ_{ij} and $\varepsilon_{ij}{}^p$, therefore, will have to be in incremental form.

At this stage it becomes necessary to consider the phenomenon of work hardening expressed through the hardening parameter k. Drucker's [31] definition of material stability postulates that during a load cycle that includes loading into the plastic range and subsequent elastic unloading, the net work performed has to be greater than zero. This plastic dissipation work is the irrecoverable part of the total work performed during the cycle.

$$d\varepsilon_{ij}{}^p \, d\sigma_{ij} \geq 0 \tag{5.14}$$

From Eqs. 5.13 and 5.14 the following conclusion can be reached.

$$d\varepsilon_{ij}{}^p = \frac{\partial g}{\partial \sigma_{ij}} \, d\lambda \tag{5.15}$$

The above equation relates the plastic strain increment to the gradient of the hyperstress surface by means of the nonnegative constant $d\lambda$. $d\lambda = 0$ represents the case of neutral loading. From Eq. 5.15 it can be observed that the plastic strain increment has the same direction as the normal to the surface g in the stress space. Thus, Eq. 5.15 is also known as the normality rule.

For the case of isotropic hardening it is assumed that g is only dependent on σ_{ij} and that the hardening is characterized by an independent parameter k^2. The yield surface is now represented by

$$g(\sigma_{ij}) = k^2 \tag{5.16}$$

For the original yield surface, k is assumed to be zero and subsequent surfaces reflecting the hardening will expand in size but retain the same shape. In order to find the plastic strain increment (Eq. 5.15), a suitable function for g has to be assumed. This also then allows for the determination of $d\lambda$. For this purpose the von Mises condition Eq. 5.7 is substituted for g and together with Eq. 5.2 the following can be written.

$$g(I_2') = g(\sigma_{ij}) = 3I_2' = \frac{3}{2}\sigma_{ij}'\sigma_{ij}' \tag{5.17}$$

The differentiation can be carried out as follows considering that $\sigma_{ij}' = \sigma_{ij} - \sigma_m\delta_{ij}$.

$$\frac{\partial}{\partial\sigma_{ij}} = \frac{1}{2}\frac{\partial}{\partial\sigma_{ij}}(\sigma'_{mn}\sigma'_{mn}) = \sigma'_{mn}\frac{\partial\sigma'_{mn}}{\partial\sigma_{ij}} = \sigma'_{ij} \tag{5.18}$$

Eq. 5.15 can now be written as

$$d\varepsilon_{ij}{}^p = \sigma_{ij}'\,d\lambda \tag{5.19}$$

relating the plastic strain increment to the stress deviator. For the two-dimensional case, Prandtl [32] first derived the stress-strain relationship in incremental form considering also elastic strains. The generalization to the three-dimensional continuum was subsequently given by Reuss [33]. Eq. 5.19 is called the Prandtl-Reuss flow rule and the resulting elastic-plastic stress-strain relationship, the Prandtl-Reuss equations.

From the above derivation it is now possible to define an expression for the equivalent plastic strain increment much in the same way as for the equivalent stress. Stating that the plastic work in the uniaxial case represented by the equivalent stress and equivalent plastic strain to the plastic work in the general case, the following holds.

$$dW^p = \sigma_e \, d\varepsilon_e^{\,p} \tag{5.20}$$

$$dW^p = \sigma_{ij} \, d\varepsilon_{ij}^{\,p} \tag{5.21}$$

In Eq. 5.21 σ_{ij} can be replaced by σ_{ij}' because the spherical part in σ_{ij} does not contribute to the plastic work. By substituting Eq. 5.19 into Eq. 5.21 and making use of Eqs. 5.17 and 5.19, the following expression is obtained.

$$dW^p = \sigma_{ij}' \, d\varepsilon_{ij}^{\,p} = \sigma_{ij}'\sigma_{ij}' \, d\lambda = 2I_2' d\lambda = \frac{2}{3} \sigma_e^{\,2} \, d\lambda \tag{5.22}$$

By setting the two plastic work expressions (Eq. 5.21 and Eq. 5.22) equal to each other, an expression for $d\lambda$ can be found.

$$d\lambda = \frac{3}{2} \frac{d\varepsilon_e^{\,p}}{\sigma_e} \tag{5.23}$$

From Eq. 5.22 and Eq. 5.12 together with Eq. 5.20 it follows that

$$\sigma_{ij}' \, d\varepsilon_{ij}^{\,p} = \frac{1}{d\lambda} \, d\varepsilon_{ij}^{\,p} \, d\varepsilon_{ij}^{\,p} = \sigma_e \, d\varepsilon_e^{\,p}$$

Solving for $d\varepsilon_e^{\,p}$ and substituting Eq. 5.23 for $d\lambda$, the desired expression for the equivalent plastic strain is obtained.

$$d\varepsilon_e^{\,p} = \left(\frac{2}{3} \, d\varepsilon_{ij}^{\,p} \, d\varepsilon_{ij}^{\,p}\right)^{\frac{1}{2}} \tag{5.24}$$

With Eq. 5.24 and Eq. 5.9 the uniaxial state is now related to the general one through the equivalent stress and the equivalent increment of plastic strain expressed in terms of the general components of stress and increments of strain.

The expression for $d\varepsilon_e{}^p$ (Eq. 5.24) can be related to the uniaxial stress-plastic strain diagram given in Fig. 5.1 by substituting for the slope H'.

$$H' = \frac{d\sigma_e}{d\varepsilon_e{}^p} \tag{5.25}$$

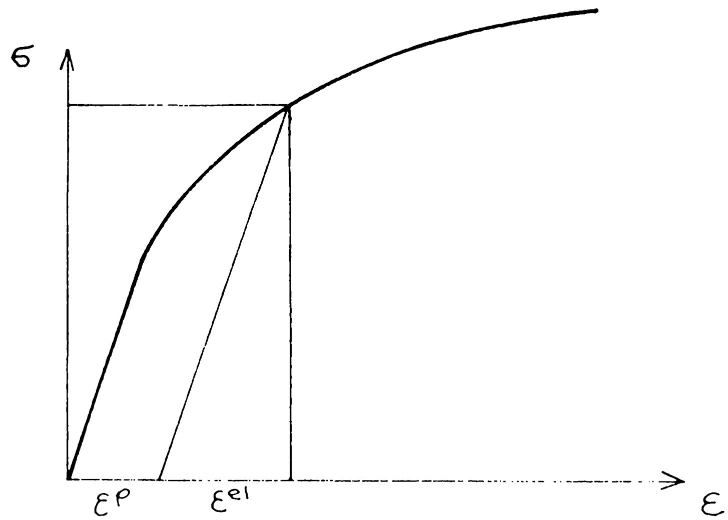
$$d\lambda = \frac{3}{2} \frac{d\sigma_e}{H'\sigma_e} \tag{5.26}$$

The Prandtl-Reuss equations giving the strain as a function of the stresses in incremental form for an isotropic strain hardening material are now expressed as follows.

$$\begin{aligned} d\varepsilon_{ij}{}' &= d\varepsilon_{ij}{}'^{e} + d\varepsilon_{ij}{}^{p} = \frac{d\sigma_{ij}{}'}{2G} + \frac{3}{2}\frac{\sigma_{ij}{}'}{H'\sigma_e} d\sigma_e \\ d\varepsilon_{ii} &= \frac{1-2\nu}{E} d\sigma_{ii} \end{aligned} \tag{5.27}$$

The first equation representing the deviator part of the strain tensor is valid only for the case of loading or $d\sigma_e > 0$. For neutral loading or unloading the plastic strain increment is zero and has to be deleted. In this case Eq. 5.25 and Eq. 5.26 reflect only elastic changes, with the second equation representing the spherical part of the strain tensor that is not affected by plastic flow.

a) Stress – total strain diagram



b) Stress – plastic strain diagram

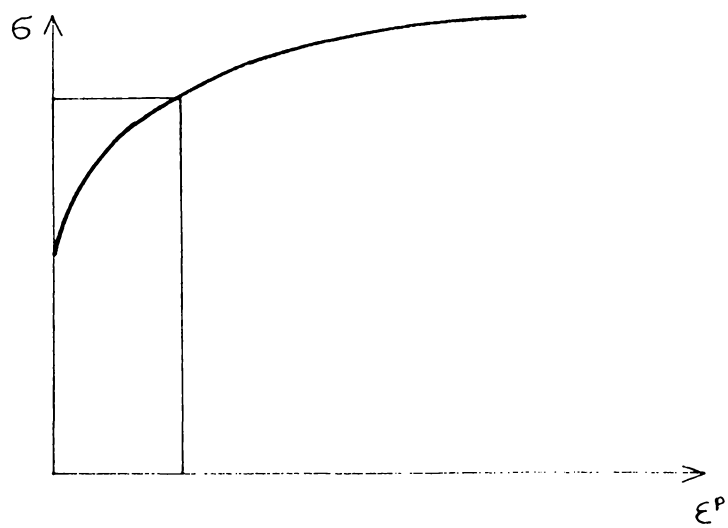


FIG. 5.1. The uniaxial stress–strain diagrams

C. The Elastic-Plastic Response in Finite-Element Formulation

The Prandtl-Reuss equations (Eqs. 5.27) given in the preceding section expressed the incremental components of strain in terms of the stress increment. In a finite-element solution, however, it is desirable to find the inverse relationship. The resulting elastic-plastic stress-strain relationship was first developed by Yamada et al. [10].

The total strain increment is given as the sum of the elastic and the plastic increment.

$$d\{\varepsilon\} = d\{\varepsilon^e\} + d\{\varepsilon^p\} \tag{5.28}$$

$d\{\varepsilon^e\}$ is related to the stresses through the elasticity matrix [D].

$$d\{\varepsilon^e\} = [D^{-1}]d\{\sigma\} \tag{5.29}$$

$d\{\varepsilon^p\}$ is taken from Eq. 5.19.

$$d\{\varepsilon^p\} = d\lambda\ \{\sigma'\} \tag{5.30}$$

At this point it has to be noted that for $d\{\varepsilon^p\}$ the "engineering strains" are used, where the shear strains are twice the magnitude of the ones used in tensor notation. For this reason, the stress deviator in matrix notation has the form

$$\{\sigma'\}^T = [\sigma_x', \sigma_y', \sigma_z', 2\sigma_{xy}, 2\sigma_{yz}, 2\sigma_{zx}] \tag{5.31}$$

The yield condition is defined as

$$g(\{\sigma'\}, k) = 0 \tag{5.32}$$

In differentiated form it can be expressed as

$$dg = \frac{\partial g}{\partial\{\sigma\}}^T d\{\sigma\} + \frac{\partial g}{\partial k}\, dk = 0 \tag{5.33}$$

By replacing the second term expressing the hardening characteristics with $d\lambda$ and an as yet unknown factor A, and substituting $\{\sigma'\}$ for $\frac{\partial g}{\partial\{\sigma\}}$ according to Eq. 5.18, the following is obtained.

$$\{\sigma'\}^T \, d\{\sigma\} - A \, d\lambda = 0 \tag{5.34}$$

Eq. 5.28 combined with Eqs. 5.29 and 5.30 and Eq. 5.34 now forms a set of two equations in the two unknowns $d\lambda$ and $d\{\sigma\}$.

$$\begin{aligned} d\{\varepsilon\} &= [D^{-1}] \, d\{\sigma\} + \{\sigma'\} \, d\lambda \\ 0 &= \{\sigma'\}^T \, d\{\sigma\} - A \, d\lambda \end{aligned} \tag{5.35}$$

Premultiplying the first equation with [D], $d\{\sigma\}$ can be expressed

$$d\{\sigma\} = [D] \, d\{\varepsilon\} - [D]\{\sigma'\} \, d\lambda \tag{5.36}$$

Substituted into the second equation we can write

$$0 = \{\sigma'\}^T[D] \, d\{\varepsilon\} - \{\sigma'\}^T[D]\{\sigma'\} \, d\lambda - A d\lambda$$

solved for $d\lambda$

$$d\lambda = \{\sigma'\}^T[D] \, d\{\varepsilon\} \, (A+\{\sigma'\}^T[D]\{\sigma'\})^{-1} \tag{5.37}$$

and substituted into Eq. 5.36, $d\{\sigma\}$ is obtained in the following form.

$$d\{\sigma\} = [D]-[D]\{\sigma'\}\{\sigma'\}^T[D](A+\{\sigma'\}^T[D]\{\sigma'\})^{-1} \quad d\{\varepsilon\} \tag{5.38}$$

$$d\{\sigma\} = [D]-[D_p] \quad d\{\varepsilon\} \tag{5.39}$$

where

$$[D_p] = \frac{[D]\{\sigma'\}\{\sigma'\}^T[D]}{A+\{\sigma'\}^T[D]\{\sigma'\}} \, .$$

The true stress increment in Eq. 5.39 is expressed as the difference between the elastic stress increment given by $[D] \, d\{\varepsilon\}$ and another stress increment given by $[D_p] \, d\{\varepsilon\}$.

As noted before (Eq. 5.23), $d\lambda$ can be related to the equivalent plastic strain increment and to the slope H' of the uniaxial stress-plastic strain curve (Eq. 5.26).

$$d\lambda = \frac{3}{2}\frac{d\varepsilon_e^{\,p}}{\sigma_e} = \frac{3}{2}\frac{d\sigma_e}{H'\sigma_e} \qquad (5.40)$$

The parameter A now remains to be determined. From the second of Eqs. 5.35 and Eq. 5.40, A can be expressed as

$$A = \frac{1}{d\lambda}\{\sigma'\}^T d\{\sigma\} = \frac{2H'}{3}\frac{\sigma_e}{d\sigma_e}\{\sigma'\}^T\, d\{\sigma\} \qquad (5.41)$$

From Eq. 5.9, through differentiation as outlined in Eq. 5.18, $d\sigma_e$ can be found.

$$\sigma_e^{\,2} = \frac{3}{2}\sigma_{ij}'\sigma_{ij}'$$

$$d\sigma_e = \frac{3}{2\sigma_e}\frac{1}{2}\frac{\partial}{\partial\sigma_{ij}}(\sigma'_{mn}\,\sigma'_{mn})\,d\sigma_{ij} = \frac{3}{2\sigma_e}\,\sigma_{ij}'\,d\sigma_{ij}$$

This is equivalent to

$$d\sigma_e = \frac{3}{2\sigma_e}\{\sigma'\}\;d\{\sigma\} \qquad (5.42)$$

Substituted into Eq. 5.41, A is now expressed as a function of the equivalent stress and the slope H' of the uniaxial stress-plastic strain diagram.

$$A = \frac{4}{9}\sigma_e^{\,2}\,H' \qquad (5.43)$$

VI. THE NONLINEAR SOLUTION PROCEDURE

While formulating the theory of the finite-element method (Chapter III), it became apparent that the linear force-displacement equations (Eq. 3.10) could be obtained only by assuming a linear strain-displacement relationship (Eq. 3.2) and linear elasticity (Eq. 3.4). Therefore, two of the most important sources of nonlinearity are already exposed. A third one is recognized as time or state dependency of any of the problem parameters. The main causes for nonlinearity can be classified in the following fashion:

a) Geometrical nonlinearities. The linear strain-displacement relationship presuming small displacements becomes invalid whenever large deformations are encountered.

b) Nonlinear constitutive relationship. Here, one can differentiate between a unique stress-strain relation as in the case of nonlinear elasticity, or an incremental relation customarily used for the case of plasticity.

c) Time or state dependency. Load conditions or material properties are allowed to change with time, or, for instance, temperature. If only the constitutive relationship is affected, this class becomes similar to class b).

In reality, the nonlinear behavior can also be caused by any combination of a), b), and c).

A. Solution Methods

For the solution of the nonlinear problem there are basically two different equilibrium strategies available. The first one is a piecewise linear, or step-by-step approach. The second one is an iterative procedure using a constant stiffness matrix.

In the first class of solutions, the aim is to represent the nonlinear load-displacement curve by a number of linear segments (Fig. 6.1). With a reasonably small stepsize, the deviations from the true nonlinear response should be small. This solution approach is also called the "tangent stiffness method." With every step, a new stiffness matrix, reflecting the changing constitutive relationship or nonlinear strain-displacement relations, has to be established and a new linear problem solved. For larger problems where the generation of the element stiffness matrices and assembly of the total stiffness matrix represent a sizeable part of the computational effort, the tangent stiffness method may no longer be feasible.

The obvious drawback of recomputing the total stiffness matrix and its complete decomposition during each step is avoided in the second class of solutions where an iterative procedure is used rather than a direct approach as described above. Often the total stiffness matrix is kept the same, thus the name "constant stiffness method." The stiffness matrix can be kept in the reduced form, and for every iteration only the modification of the new load vector and the backsubstitution has to be performed. The two most widely used iterative processes are the initial stress and the initial strain methods. For a one degree-of-freedom

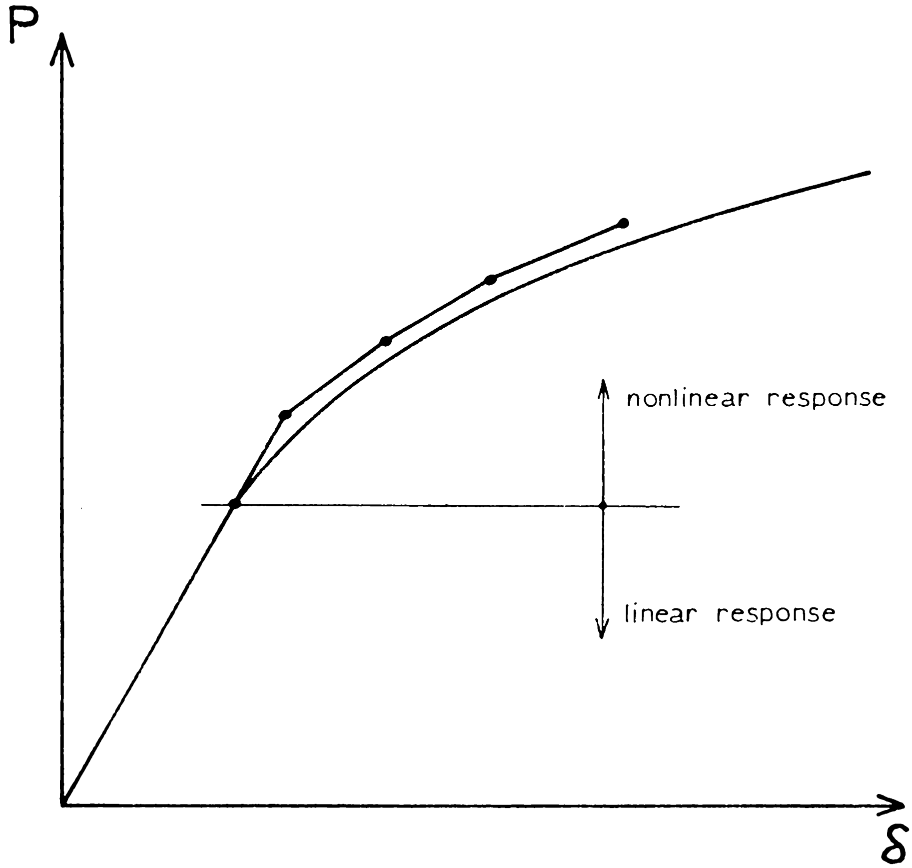


FIG. 6.1. Step-by-step solution

system, where the load-displacement curve is identical to the stress-strain curve, the initial stress method is represented in Fig. 6.2, and the initial strain method in Fig. 6.3. In the initial stress method it is the difference between the elastic stress σ^{el} and the true elastic-plastic stress σ^{e-p} that has to be introduced as initial stress in order to maintain equilibrium. In the initial strain approach, the difference between the true strain ε^{e-p} and the elastic strain ε^{el} is treated as an initial strain condition, again to satisfy equilibrium. In this investigation only the initial stress method will be discussed.

B. The Initial Stress Method

In order to describe the steps involved in the initial stress approach for a given load increment, it is assumed that the total stresses $\{\sigma\}$, strains $\{\varepsilon\}$, and the equivalent plastic strain ε_e^p at the start of the increment are known throughout the region. This information is needed to check the yield condition (Eq. 5.8) and also to find the elastic-plastic increment of stress (Eq. 5.37). A typical iterative step is illustrated in a flow-diagram in Fig. 6.4.

The process is started either with a new increment of the external load, in which case the first elastic problem is solved directly, or with a set of initial stresses $\{\sigma_i^{init}\}$.

The equivalent nodal forces $\{\Psi\}$ representing these initial stresses are then found with the aid of Eq. 3.18.

With the original stiffness matrix [k] in decomposed form and the loading $\{\Psi\}$, the corresponding displacements $\{\Delta\delta_i^{el}\}$ can be evaluated.

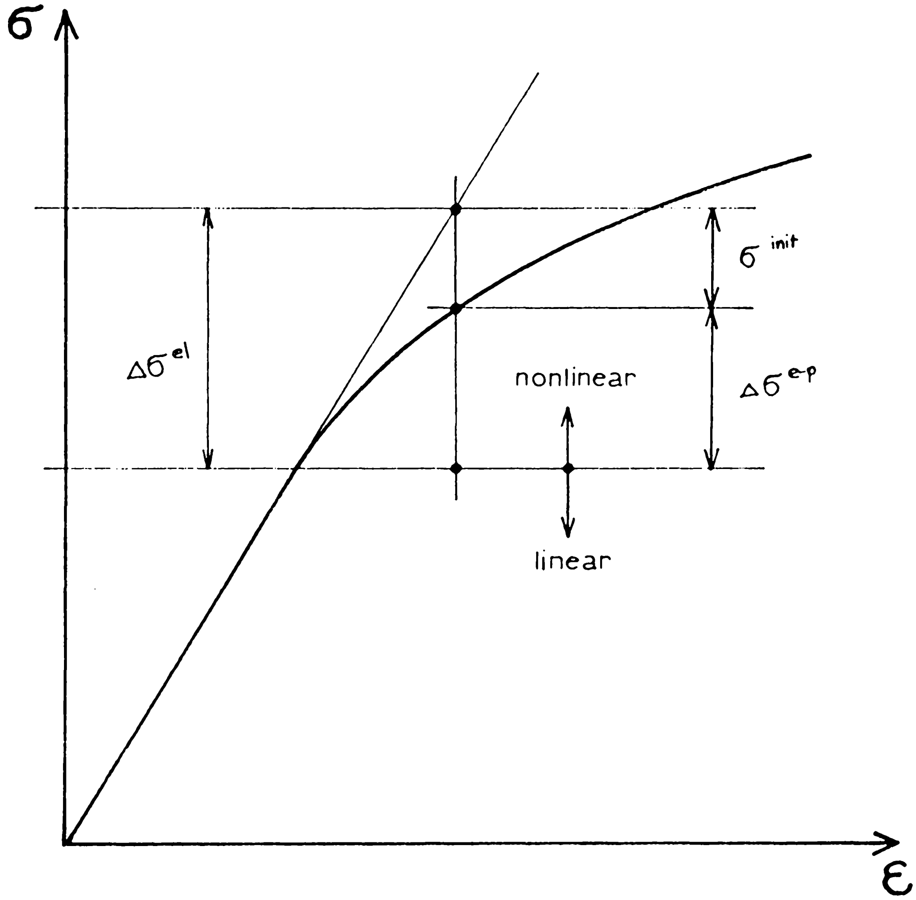


FIG. 6.2. Initial stress solution

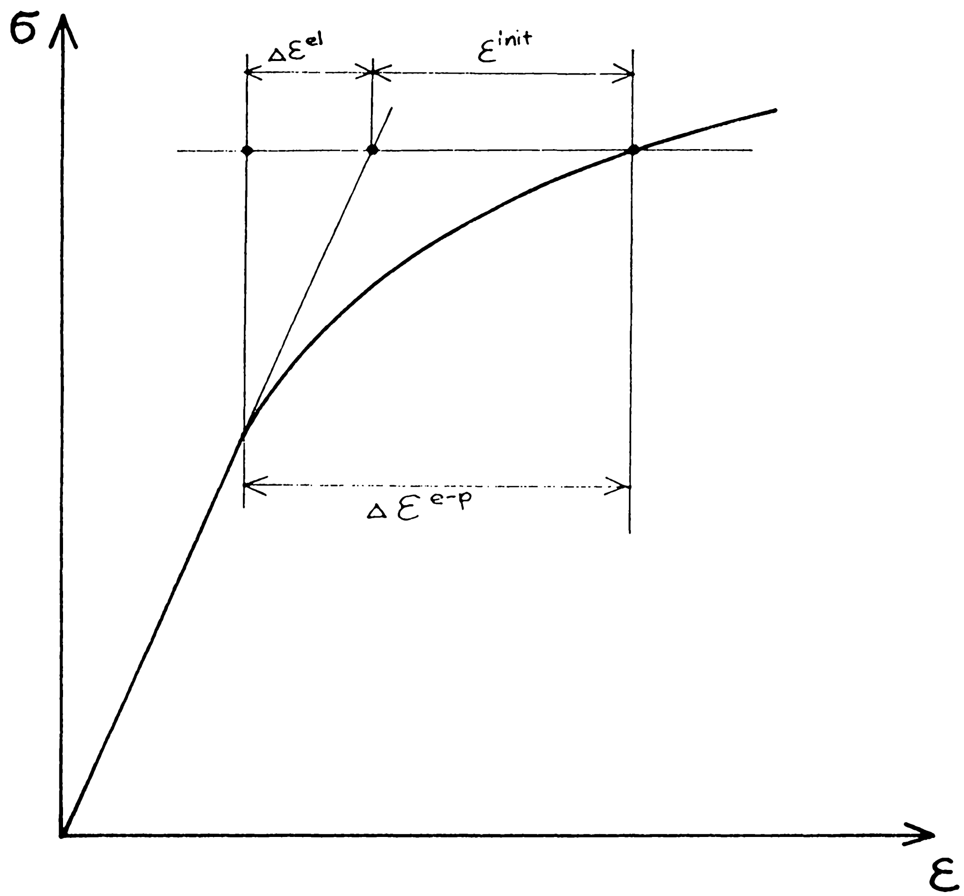


FIG. 6.3. Initial strain solution

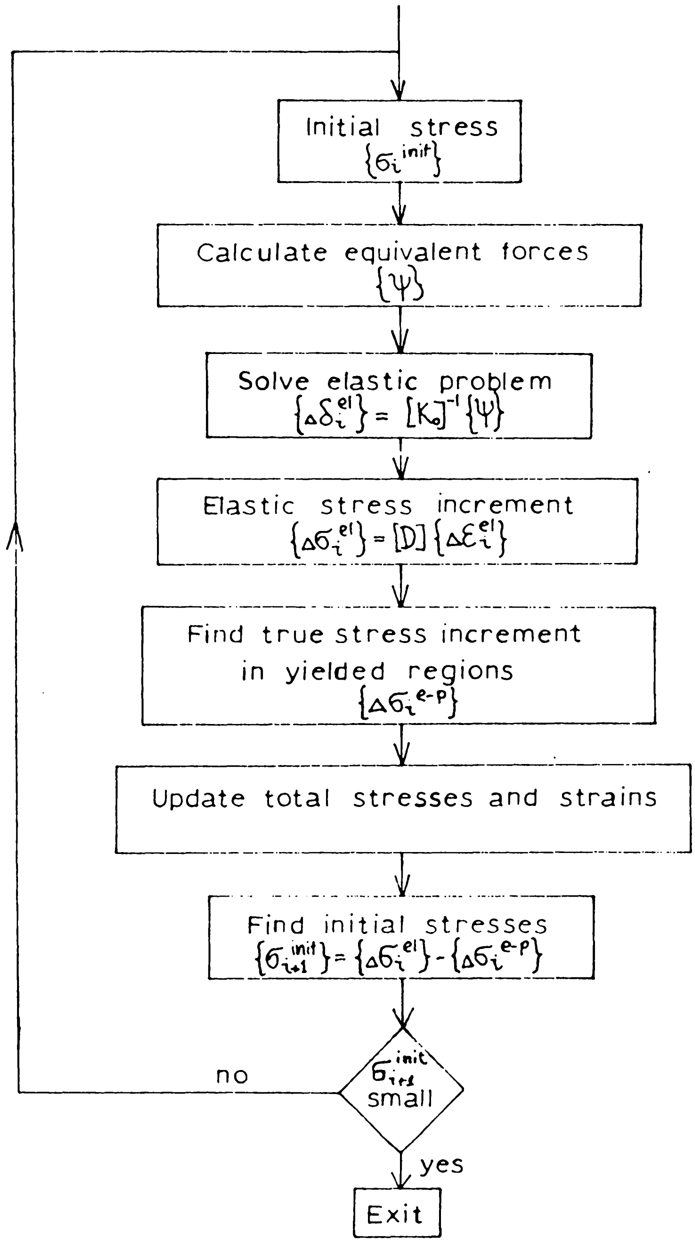


FIG. 6.4. Initial stress flow diagram

The elastic stresses $\{\Delta\sigma_i^{el}\}$ are determined by means of the linear elasticity matrix [D].

The evaluation of the correct elastic-plastic stress increment $\{\Delta\sigma_i^{e-p}\}$ from the given strain increment $\{\Delta\varepsilon_i^{el}\}$ has to follow the rules that were derived in general terms in Chapter V. The specific steps of finding the elastic-plastic stress increment are outlined in the following section of this chapter.

With the correct stress increment known, it can now be added to the total stresses. Likewise, the increment of strain $\{\Delta\varepsilon_i^{el}\}$ is added to the total strains. The equivalent plastic strain increment of the current iterative step $\Delta\varepsilon_e^{p}$ is also found during the evaluation of the elastic-plastic stress.

The difference between $\{\Delta\sigma_i^{el}\}$ and $\{\Delta\sigma_i^{e-p}\}$ represents the new initial stress $\{\sigma_{i+1}^{init}\}$ used to start the next iteration.

The above procedure is repeated until the initial stresses $\{\Delta\sigma_i^{init}\}$ become negligible, or in other words, until the iterative process has converged within a desired accuracy.

C. The Evaluation of the Elastic-Plastic Stresses

During every iterative step, wherever plastic flow is encountered, the appropriate elastic-plastic stress increment $\{\Delta\sigma_i^{e-p}\}$ has to be evaluated for every strain increment $\{\Delta\varepsilon_i^{el}\}$.

If for a given point plastic behavior was initiated or already present during the preceding iterative step, the response for the current step will be determined as follows:

1) If the yield function F_1 (Eq. 5.32) is greater than zero,

$$F_1(\{\sigma_{tot}\}+\{\Delta\sigma_i^{el}\},\ \sigma_Y) > 0 \tag{6.1}$$

loading is taking place and the whole increment will be governed by elastic-plastic behavior.

2) If F_1 is smaller than zero, unloading occurs and the evaluation of the elastic-plastic stresses is omitted. The response for the current iterative step will be purely elastic.

An additional complication occurs when the yield condition $F_1 > 0$ and ε_u^p is zero. This means that for the point under consideration elastic-plastic behavior is initiated during the present iterative step. In order to find the load level during the increment where the elastic response changes to elastic-plastic, a linear interpolation is carried out. The yield condition for the start of the step is expressed as follows

$$F_0(\{\sigma_{tot}\},\sigma_Y) < 0 \tag{6.2}$$

and the yield condition with $\{\Delta\sigma_i^{el}\}$ added to the total stresses as in Eq. 6.1. The interpolation factor is defined as

$$r = -\frac{F_0}{F_1 - F_0} \tag{6.3}$$

In such a case part of the increment will be in the elastic range. The total stress is raised to the yield level by adding $r\{\Delta\sigma_i^{el}\}$. The initial stress $\{\sigma_i^{init}\}$ then has to be determined from the $(1-r)\{\Delta\varepsilon_i^{el}\}$ part of the elastic strain increment of the iterative step.

The steps taken for finding the initial stress $\{\sigma_i^{init}\}$ at a given point are outlined below and illustrated in Fig. 6.5. From Eq. 5.39, which is repeated below, the true stress increment $d\{\sigma\}$ is found from the elastic increment and the initial stress.

$$d\{\sigma\} = ([D]-[D_p])\ d\{\varepsilon\} \qquad (5.39)\ \text{rep.}$$

The initial stress $\{\sigma_i^{init}\}$ corresponds to the expression $[D_p]\ d\{\varepsilon\}$.

From Eq. 5.38, $[D_p]$ can be expressed as

$$[D_p] = [D]\{\sigma'\}\{\sigma'\}^T[D]\ (A+\{\sigma'\}^T[D]\{\sigma'\})^{-1} \qquad (6.4)$$

Comparing Eq. 6.4 to Eq. 5.37, the following equation holds.

$$d\{\sigma_i^{init}\} = [D_p]\ d\{\varepsilon\} = [D]\{\sigma'\}\ d\lambda \qquad (6.5)$$

The only quantity in this expression for the initial stresses that has yet to be determined is $d\lambda$. With Eqs. 5.37 and 5.41 combined, $d\lambda$ will be expressed entirely in terms of known quantities.

$$d\lambda = \{\sigma'\}^T[D]\ d\{\varepsilon\}\ (\tfrac{4}{9}\ \sigma_e^{\,2}H' + \{\sigma'\}^T[D]\{\sigma'\})^{-1} \qquad (6.6)$$

Here it should be pointed out that for practical purposes the differential $d\{\varepsilon\}$ is replaced by the increment $\{\Delta\varepsilon\}$.

The magnitude of H' representing the slope of the equivalent stress-equivalent plastic strain curve (Fig. 5.1) is a function of the accumulated equivalent plastic strain $\varepsilon_e^{\,p}$. Therefore, during every iterative step the increment $d\varepsilon_e^{\,p}$ has to be evaluated according to Eq. 5.38 and added to the total uniaxial plastic strain.

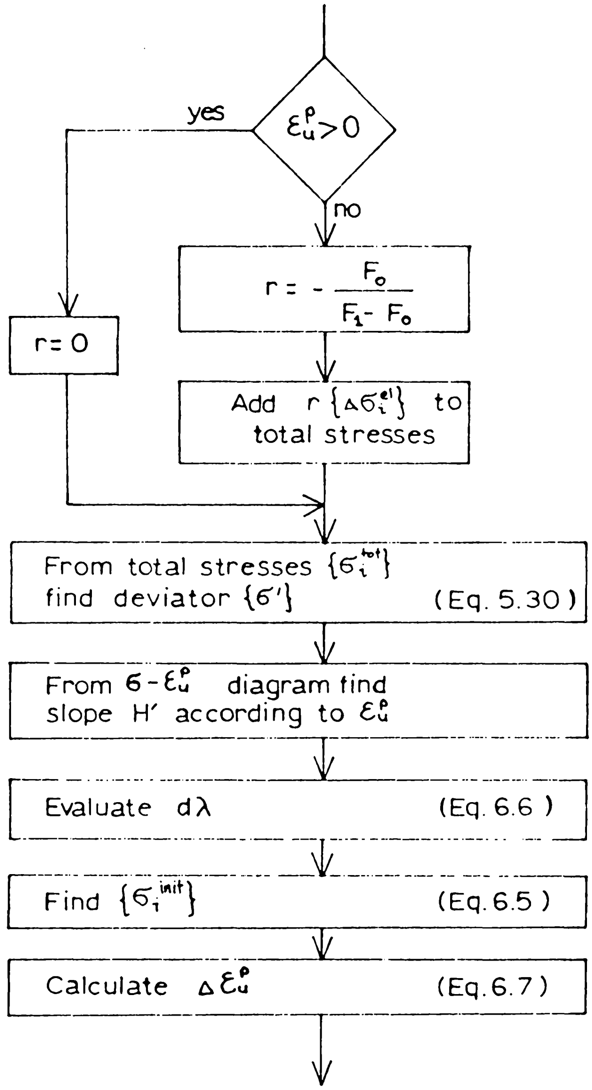


FIG. 6.5. Evaluation of initial stresses

$$d\lambda = \frac{3}{2} \frac{d\varepsilon_e^{\ p}}{\sigma_e} \qquad (5.40) \text{ rep.}$$

$$d\varepsilon_e^{\ p} = \frac{2}{3} \frac{d\lambda}{\sigma_e} \qquad (6.7)$$

From the shape functions (Eqs. 4.3) relating the displacements at an arbitrary point to the nodal displacements of an element, it becomes clear that the stresses

$$\{\sigma\} = [D][B]\{\delta^e\} - [D]\{\varepsilon_0\} \qquad (4.26) \text{ rep.}$$

can be evaluated at any desired location, with the [B] matrix dependent on its coordinates. From the flow-diagram of Fig. 6.4 it can be seen that during every iterative step, equivalent forces $\{\Psi\}$ are calculated to equilibrate the initial stresses $\{\sigma_i^{\ init}\}$.

This load vector is obtained through summation of the individual element nodal forces that were given previously

$$\{F\}_{\sigma_o}^{\ e} = \int_V [B]^T\{\sigma_o\}\, dV \qquad (3.18) \text{ rep.}$$

For the evaluation of the above integral the same integration rule described in Chapter IV, Section D, is employed. This means that in order to find the equivalent $\{\Psi\}$ the initial stresses at the integration points have to be known. Therefore, the 14 integration points lend themselves conveniently as locations where total stresses, total elastic and plastic strains, and the hardening characteristics are monitored.

VII. CONVERGENCE AND ACCELERATION

The convergence of the iterative scheme as reflected in the diminishing residual force vector $\{\Psi\}$ and an acceleration method are discussed in this chapter. A new scheme accelerating the displacements throughout the domain by the same factor, and thereby preserving equilibrium is presented. This procedure is named uniform acceleration.

A. Convergence

With the aid of the iterative scheme described in the preceding chapter, the unbalanced forces $\{\Psi\}$ due to the initial stresses $\{\sigma^{init}\}$ become negligibly small. As a measure for their magnitude, the norm of the vector $||\Psi||$ is used. Most frequently applied are the Euclidean norm (Eq. 7.1) and the absolute value of the largest component of the vector (Eq. 7.2).

$$||X|| = (|X_1|^2 + |X_2|^2 + \ldots + |X_n|^2)^{1/2} \tag{7.1}$$

$$||X|| = \max_i |X_i| \tag{7.2}$$

For this investigation the maximum absolute value of the largest component of the residual force vector was used to monitor the convergence of the iterative procedure. As the criterion for convergence the following was used.

$$||\Psi|| = \max_i |\Psi_i| \leq \varepsilon = .05 \tag{7.3}$$

If convergence is not achieved for a given number of iterations, the conclusion is drawn that the load bearing capacity of the structure is exhausted. For the case where $||\Psi||$ is not reduced appreciably during a number of iterative steps, it is clear that an increase in the displacements is observed without an increase in the loads. For this reason the analysis is terminated and collapse of the structure is assumed.

B. Uniform Acceleration

Any acceleration scheme aims to adapt the slope of the linear force-displacement response more towards the true elastic-plastic response. For a one degree-of-freedom system Fig. 7.1 shows qualitatively how the number of iterations is reduced. Nayak and Zienkiewicz [18] derived the alpha-constant acceleration scheme which attempts to predict the displacement increment for the next iterative step based on the current and the preceding increment. Fig. 7.2 illustrates the evaluation of the individual acceleration factors. If δ_1 and δ_2 are known, then, assuming a constant slope of the response curve, δ_3 is predicted as follows.

$$\frac{\delta_1}{\delta_2} = \frac{\delta_2}{\delta_3}$$

$$\delta_3 = \delta_2 \frac{\delta_2}{\delta_1} \tag{7.4}$$

Now, the quantity δ_3 is added to δ_2 and in such a fashion the accelerated displacement δ_2' is expressed as

$$\delta_2' = \delta_2 + \delta_3 = \delta_2(1 + \frac{\delta_2}{\delta_1}) = \alpha\ \delta_2 \tag{7.5}$$

where the acceleration factor α is

$$\alpha = 1 + \frac{\delta_2}{\delta_1} \tag{7.6}$$

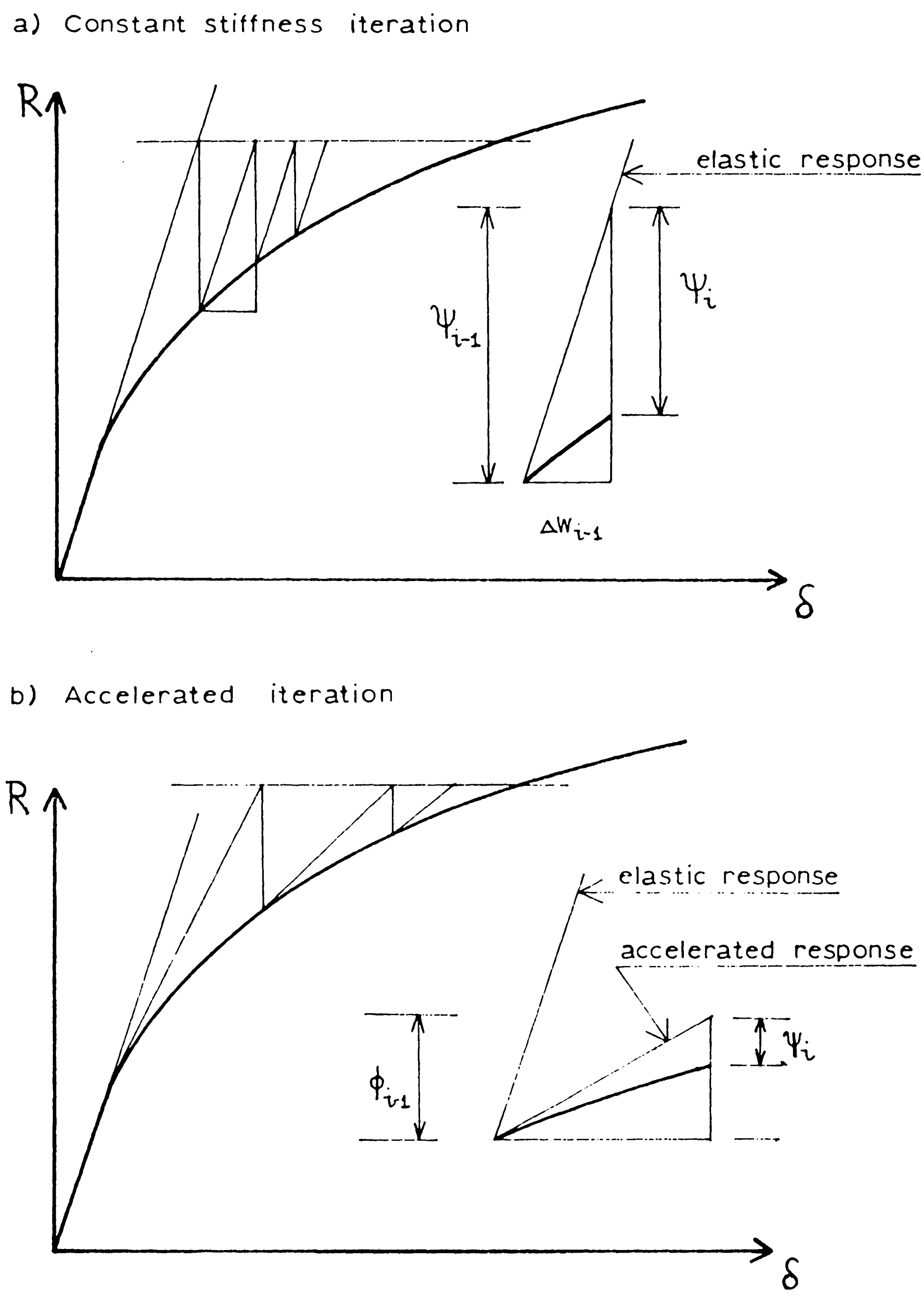
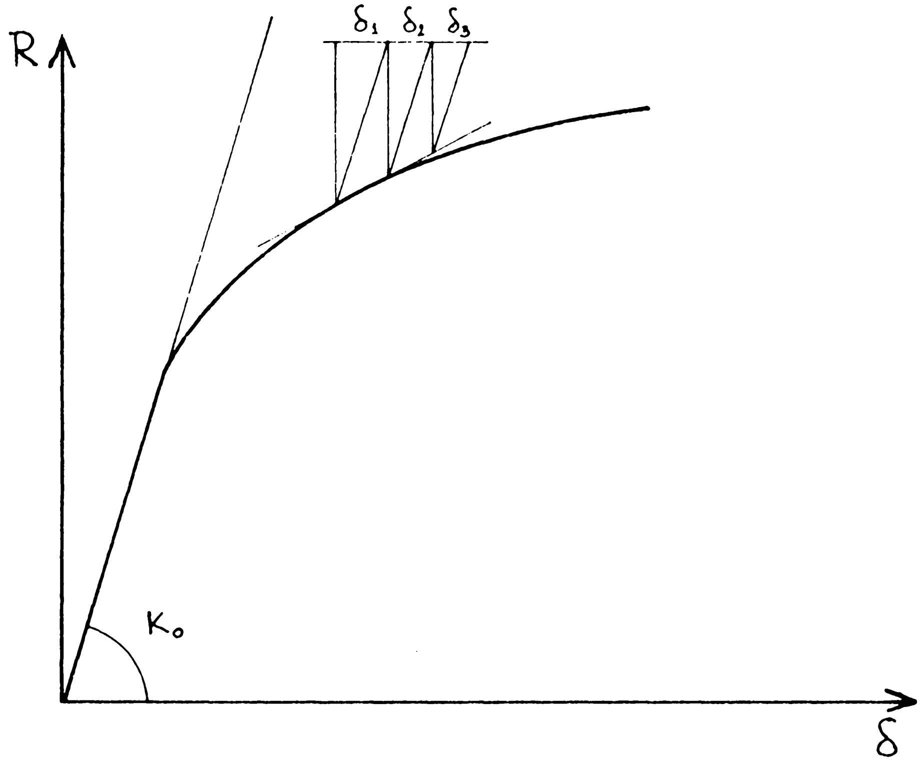


FIG. 7.1. Initial stress iteration schemes



Because of similarity of triangles :

$$\frac{\delta_1}{\delta_2} = \frac{\delta_2}{\delta_3} \quad , \qquad \delta_3 = \delta_2 \frac{\delta_2}{\delta_1}$$

$$\delta_2' = \delta_2 + \delta_3 = \delta_2 \left(1 + \frac{\delta_2}{\delta_1}\right) = \alpha \, \delta_2$$

FIG. 7.2. Acceleration factor α

This factor α is always between 1.0 and 2.0. If no acceleration is taking place at all α is equal to 1.0 and if the δ-R curve is approaching the slope zero, the α factor approaches the maximum value of 2.0.

The acceleration factor α is computed individually for every degree-of-freedom of the system. The displacement vector obtained from the linear stiffness matrix $[k_0]$ then is multiplied by the $[\alpha]$ matrix which contains the above mentioned constants on its diagonal. The rest of the components of $[\alpha]$ are zero. The accelerated displacement vector is then expressed as follows.

$$\{\Delta w_i'\} = [\alpha_{i-1}]\,\{\Delta w_i\} \tag{7.7}$$

The initial stresses are found for the accelerated displacements $\{\Delta w_i'\}$ and the forces $\{\Psi\}$ equilibrating $\{\sigma^{init}\}$ are used for the next linear solution,

$$\{\Delta\bar{w}_i\} = [k_0]^{-1}\{\Psi_i\} \tag{7.8}$$

which gives rise to a new set of acceleration factors $[\alpha_i]$. A better estimate for the displacement increment $\{\Delta w_i''\} = [\alpha_i]\{\Delta w_i\}$ is used to find the true increments of stress which are then used to update the total stresses.

The residual forces $\{\phi_i\}$ needed to reestablish equilibrium are then found as follows.

$$\{\phi_i\} = \int_V [B]^T\{\sigma_i^{tot}\}\,dV - \{R\} \tag{7.9}$$

where $\{R\}$ represents the total external load. A new displacement increment $\{\Delta w_{i+1}\}$ results from

$$\{\Delta w_{i+1}\} = -[k_0]^{-1}\{\phi_i\} \tag{7.10}$$

and with the aid of $[\alpha_i]$ they are accelerated by application of Eq. 7.7.

In performing a typical iterative step (Eqs. 7.7 - 7.10) there are two stages where a numerical integration is required. The first one is the evaluation of $\{\Psi_i\}$ from the initial stresses, which requires integration over the elastic-plastic region only. For the calculation of $\{\phi_i\}$, however, the integration has to be carried out over the whole domain of the analysis. The reason for this is that this type of acceleration destroys the equilibrium, and therefore, it has to be reestablished by finding the residual forces $\{\phi_i\}$. The individually accelerated displacements produce a stress state that is not in balance with the residual forces of the previous iteration. Hence, the new residual forces $\{\phi_i\}$ are represented by the difference between the equivalent nodal forces, balancing the total stresses $\{\sigma^{tot}\}$ and the external forces $\{R\}$ (Eq. 7.9).

In this investigation an acceleration scheme is adopted that preserves equilibrium, and therefore, makes the evaluation of equivalent forces over the whole region unnecessary. This is achieved by applying the same acceleration factor γ to all displacements. The factor γ is found by averaging the individual factors α_i found in the same fashion as in Eq. 7.6.

$$\gamma = \frac{1}{n}\sum_{i=1}^{n} \alpha_i \tag{7.11}$$

Because this factor γ is applied to all displacements, this method is called uniform acceleration.

With the use of only one common factor the force-displacement equation can be rewritten as follows

$$\begin{aligned} \{\Delta w_i\} &= [k_0]^{-1} \{\Psi_i\} \\ \gamma\{\Delta w_i\} &= [k_0]^{-1} \gamma\{\Psi_i\} \end{aligned} \tag{7.12}$$

By multiplying both sides with the scalar quantity γ, equilibrium remains undisturbed. During the evaluation of the residual forces $\{\Psi_{i+1}\}$, however, it is important to note that the previous $\{\Psi_i\}$ was multiplied by γ. As a consequence, in the evaluation of the new residual forces from the initial stresses, the previously accelerated forces have to be considered.

$$\{\Psi_{i+1}\} = \int_V [B]^T \{\sigma_i^{\,init}\}\, dV - (\gamma-1)\{\Psi_{i-1}\} \tag{7.13}$$

A graphical representation in a one-dimensional context for the uniform acceleration scheme is given in Fig. 7.3.

The numerical integration is carried out over the domain in which elastic-plastic behavior governs. In many problems this region remains small compared to the total domain. Hence, considerable computational savings are realized for the evaluation of the residual forces.

While running several example problems, it was observed that the individual acceleration factors associated with the displacements for which the largest changes were taking place, were almost the same. For this reason the averaging of the individual acceleration factors has been restricted to those degrees-of-freedom showing the most significant changes in the displacements. For example, in the plate problem only

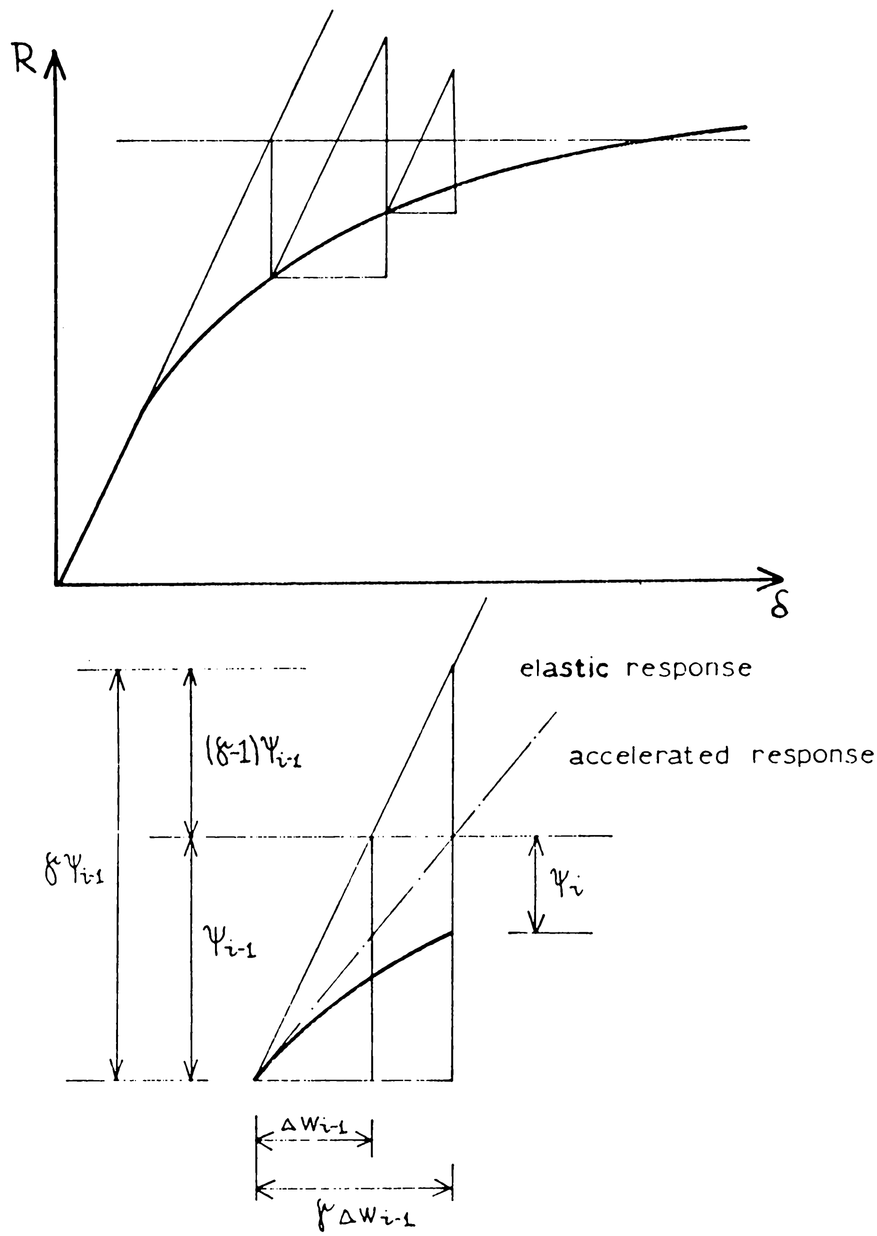


FIG. 7.3. Uniform acceleration

the degrees-of-freedom representing the vertical deflections were considered for the determination of the average acceleration factor γ. The in plane displacements for this problem did not influence the evaluation of γ. With this arrangement possibly erroneous acceleration factors due to insignificant displacement changes several magnitudes smaller, were prevented from adversely affecting the acceleration.

VIII. THE COMPUTER PROGRAM

In this chapter the major steps of the computer program used in this study are described and some special features are examined in greater detail. The total program consists of two main parts representing the linear analysis and the subsequent nonlinear problem solution. Appendix 3 contains a complete listing of the total program.

A. The Linear Analysis

The flow of the program handling the linear analysis indicated in Fig. 8.1 involves the same standard procedures as can be seen in many other finite-element programs. This linear program is designed to operate as an independent unit and furnish nodal displacements and stresses at the integration points of the elements for problems adequately represented by a linear analysis.

The modification of the total stiffness matrix accounting for boundary conditions and in some cases rotation of the coordinate system for certain nodes, usually performed after the assembly of the total matrix, is carried out in this program on the element level.

Prescribed displacement boundary conditions restricting certain degrees-of-freedom to a given value are enforced in the following manner. If for a given element the displacement δ_n is set to a value of Δ, the corresponding row and column of the stiffness matrix $[k]^e$ are set to zero. The products of the n^{th} column of the stiffness matrix with Δ are transferred to the right hand side. The n^{th} diagonal of $[k]^e$ then is set to 1.0 and the n^{th} component of $\{R\}$ is replaced with Δ. In the

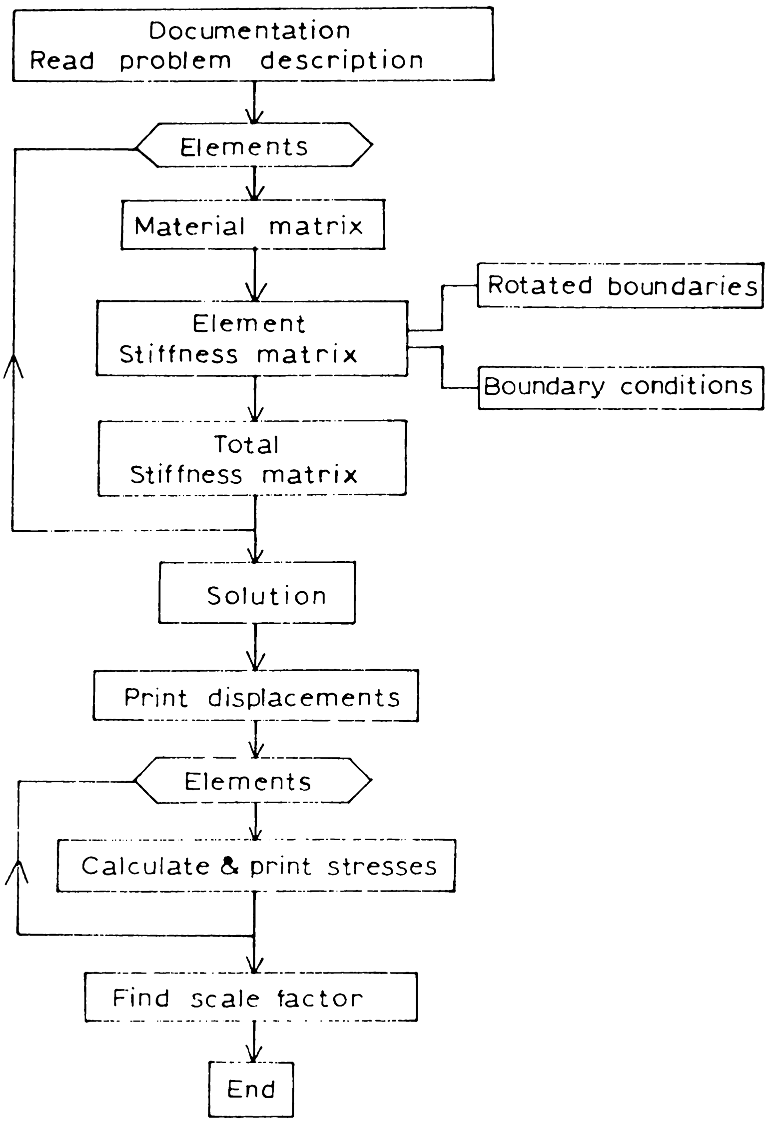


FIG. 8.1. Flowchart for linear program

original form the load-displacement relationship for an element is written as follows.

$$[k]^e\{\delta\}^e = \{R\}^e \tag{8.1}$$

In expanded form the modified set of equations can be expressed as

$$\begin{bmatrix} k_{11} & k_{12} \; \cdots & 0 & \cdots \; k_{1m} \\ k_{21} & & \vdots & \vdots \\ 0 & \cdots & 1 & 0 \\ \vdots & & & \vdots \\ k_{m1} & & 0 & k_{mm} \end{bmatrix} \begin{Bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_n \\ \vdots \\ \delta_m \end{Bmatrix} = \begin{Bmatrix} R_1 - k_{1n} \cdot \Delta \\ R_2 - k_{2n} \cdot \Delta \\ \vdots \\ \Delta \\ \vdots \\ R_m - k_{mn} \cdot \Delta \end{Bmatrix} \tag{8.2}$$

and the n^{th} equation now reads as follows

$$1.0 \cdot \delta_n = \Delta \tag{8.3}$$

Care has to be taken that every element matrix containing the n^{th} degree-of-freedom is modified in such a fashion and that all the affected load terms are corrected from their original values or their current values due to changes caused by previous element modifications.

The compilation of the total stiffness matrix then does not pose any additional problems. The usual superposition of the element stiffness matrix is carried out. However, a possible accumulation of the 1.0 values on the diagonal has to be prevented.

In some problems displacement boundary conditions that can be expressed only as linear combinations of the global coordinates have to be enforced. This is the case if the problem description takes advantage of axes of symmetry that are rotated with respect to the global x, y, and z axes, or if roller bearings moving on an inclined plane are given. If the displacement boundary conditions for any node are specified in a rotated x', y', z' system, a standard coordinate transformation relating the global forces and displacements to the local values has to be performed.

$$\{R'\}^e = [S]\{R\}^e \tag{8.4}$$

$$\{\delta'\}^e = [S]\{\delta'\}^e \tag{8.5}$$

The transformation matrix [S] is square and of the same size as the number of degrees-of-freedom of an element. It is built up of 3*3 submatrices placed on the diagonal. These submatrices represent the transformation for individual nodal points and contain the direction cosines of the rotated system (x',y',z') with respect to the global system. In theory, a different coordinate system could be adopted for every nodal point. For nodes where no transformation is required, [S] shows only diagonal terms of value 1.0. The global force-displacement relationship of an element as given in Eq. 8.1 is modified by substituting Eqs. 8.4 and 8.5 that are premultiplied with $[S]^T$.

$$[k]^e[S]^T\{\delta'\}^e = [S]^T\{R'\}^e \tag{8.6}$$

In this operation orthogonality of the transformation matrix [S] is implied, and therefore, the product of [S] with its transpose yields the identity matrix. Eq. 8.6 is now premultiplied by [S] as indicated below.

$$[S][k]^e[S]^T\{\delta'\}^e = \{R'\}^e \tag{8.7}$$

A comparison of Eq. 8.7 with the force-displacement relationship expressed in the primed coordinates

$$[k']^e\{\delta'\}^e = \{R'\}^e \tag{8.8}$$

yields the expression for the modified stiffness matrix given below.

$$[k']^e = [S][k]^e[S]^T \tag{8.9}$$

With respect to the load vector, the prescribed displacements and external loads have to be specified in the (x',y',z') system according to Eq. 8.4.

After the transformations on the element stiffness matrix and on the load vector are carried out, the modification of the load-displacement equations due to boundary conditions is performed in the same fashion as described in Eq. 8.4.

After every element stiffness matrix is evaluated and modified, if necessary, the individual coefficients are transferred into the total stiffness matrix node-by-node. The three equations associated with the given node are used from the proper storage location and the contributions of the current element are superimposed to the present values.

For the solution of the total load-displacement equations a Gauss elimination routine named USOL [34] is employed. The full title reads as follows: USOL - General Equation Solver for Symmetric Positive Definite Systems with Practically No Limit on the Number of Equations, Bandwidth, or Number of Solution Vectors. Of course, with increasing bandwidth and number of unknowns the solution time increases rapidly and particularly the transfer times to and from direct access devices become prohibitive.

From the primary unknowns, the displacements, the stresses are evaluated element-by-element. The stress calculation at the 14 integration points of each element is based on the developments of Chapter IV, Section H.

As a last step in the linear solution, the integration point with the largest equivalent stress compared to the uniaxial yield stress of the respective material is determined. The smallest ratio

$$SF = \frac{\sigma_Y}{\sigma_e} \tag{8.10}$$

represents the scale factor by which the original load has to be multiplied in order to initiate yield for at least one location of the continuum. Any additional load increment over the scaled loading will induce plastic flow and is governed by a nonlinear response.

B. The Nonlinear Analysis

The elastic-plastic response is handled by a second program which starts with a first jobstep that repeats the linear solution for the scaled loading.

A second jobstep is characterized by an outer loop with the number of individually specified load increments as parameters. Embedded in this loop is the iterative scheme. The programmed flow of the analysis is illustrated with the chart in Fig. 8.2. The evaluation of the elastic-plastic stresses and the uniform acceleration scheme have been previously described in Chapters VI and VII respectively. The evaluation of the equivalent forces balancing the initial stresses follows from Chapter IV, Section H.

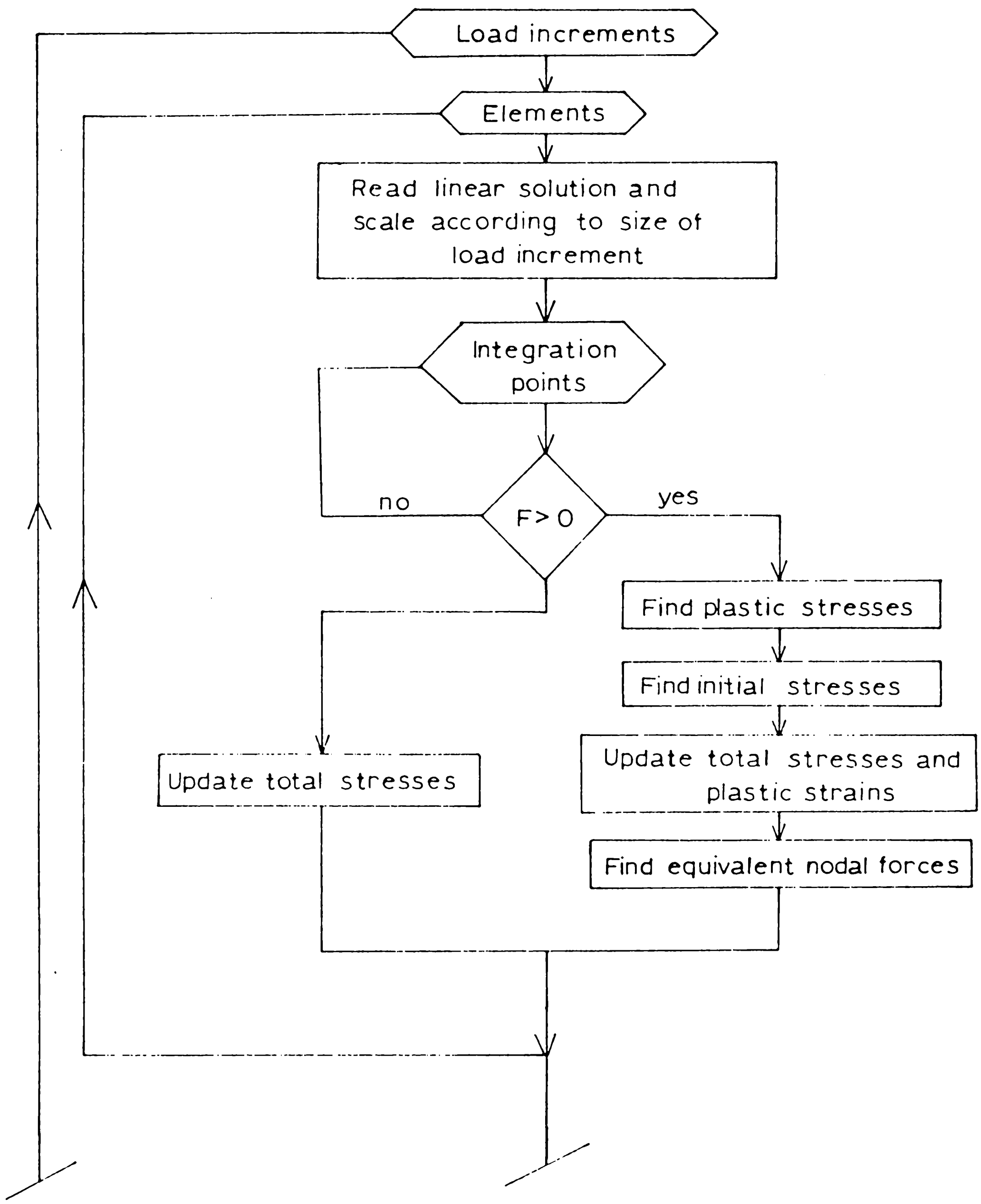


FIG. 8.2. Flow of nonlinear solution

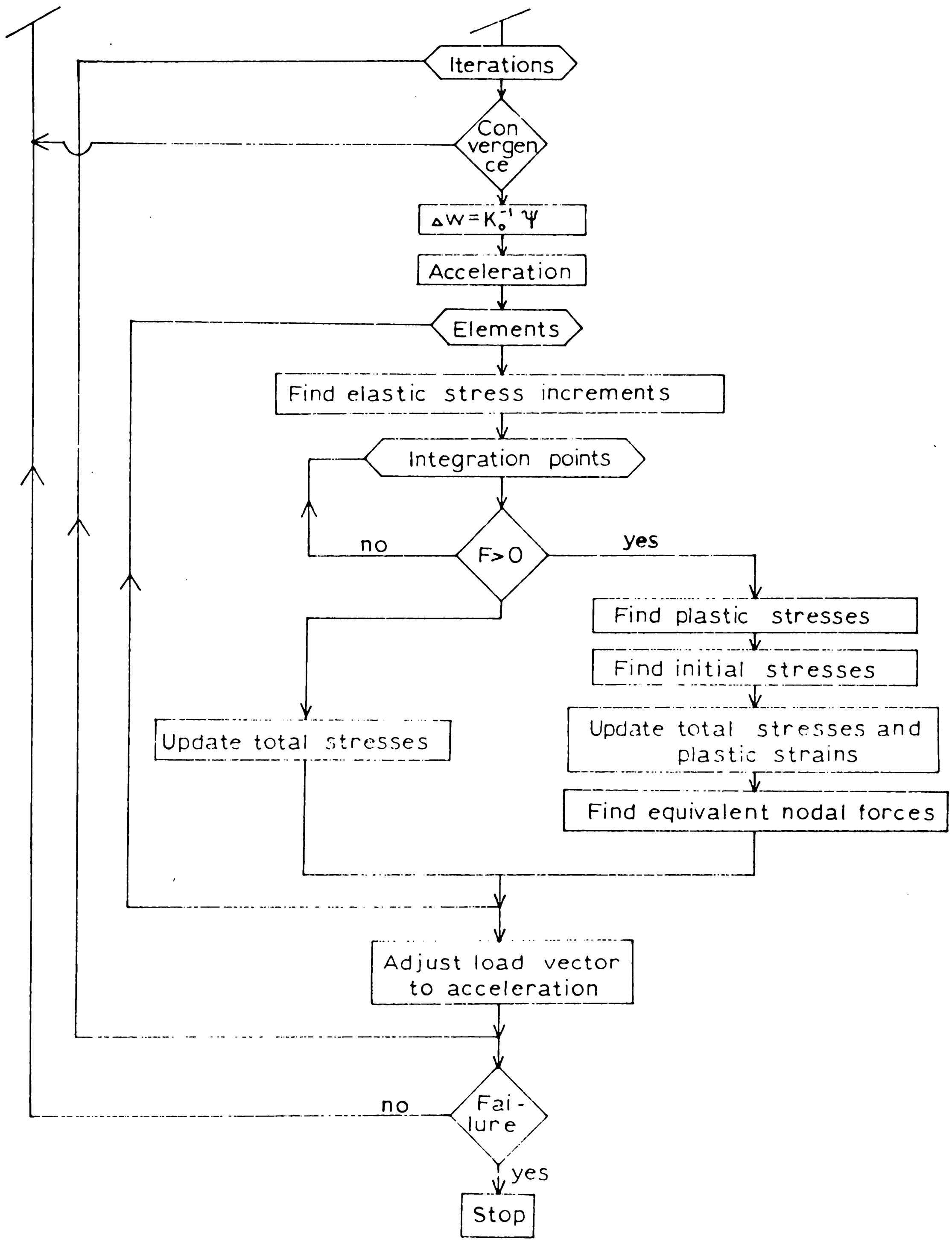

Iterations
Convergence
$\Delta W = K_0^{-1} \Psi$
Acceleration
Elements
Find elastic stress increments
Integration points
no
F>0
yes
Find plastic stresses
Find initial stresses
Update total stresses
Update total stresses and plastic strains
Find equivalent nodal forces
Adjust load vector to acceleration
no
Failure
yes
Stop

FIG. 8.2. cont'd

During every iteration, a linear solution is obtained from

$$\{\Delta w\} = [k_0]^{-1}\{\Psi\} \tag{8.11}$$

In the preceding linear analysis the subroutine USOL featured the three distinctive steps found in a Gauss elimination scheme: the decomposition of the original equations into triangular form, a forward substitution, and a backsubstitution. In decomposed form the positive definite stiffness matrix is expressed in the following manner

$$[k_0] = [L][\overline{D}][L]^T \tag{8.12}$$

where [L] represents the triangular matrix with nonzero terms below the diagonal whose components are all of the value 1.0, and $[\overline{D}]$ a diagonal matrix.

During the first linear solution [L] is generated and kept. Hence, it is available for all subsequent solutions. For convenience, the terms of $[\overline{D}]$ are stored on the diagonal of [L]. With Eq. 8.12 substituted, the overall load-displacement relationship is written as

$$[L][\overline{D}][L]^T\{\delta\} = \{R\} \tag{8.13}$$

Premultiplying both sides with $([L][\overline{D}])^{-1}$ yields

$$[L]^T\{\delta\} = ([L][\overline{D}])^{-1}\ \{R\} = \{G\} \tag{8.14}$$

and from the above it follows that

$$[L][\overline{D}]\ \{G\} = \{R\} \tag{8.15}$$

The matrix product $[L][\overline{D}]$ is formed by multiplying the columns of [L] with the appropriate term of $[\overline{D}]$.

The modified load vector {G} can now be found from Eq. 8.15 through a forward substitution process. From Eq. 8.14 the desired displacement vector $\{\delta\}$ is obtained by means of backsubstitution.

IX. APPLICATIONS

Three different nonlinear material problems that were successfully solved with the described analysis are discussed in this chapter. The first one, a thick cylinder under internal pressure was chosen to verify the present analysis on a well documented example. For the second problem of a thin plate under transverse loading, a previous special purpose finite-element analysis provided a valuable check. The third problem, the stress distribution in the vicinity of a free surface of a fiber composite under a temperature loading, has not been documented before.

A. Thick Walled Cylinder

The problem of the long thick walled cylinder under internal pressure lends itself to a two-dimensional plane strain formulation. An analytical solution was given in closed form by Reckling [35] and a finite-element analysis was carried out by Nayak and Zienkiewicz [12].

Here, a three-dimensional analysis on a thirty degree section of the cylinder is performed. The finite-element idealization is given in Fig. 9.1. The same solution accuracy could have been obtained with only three elements representing a fifteen degree slice. The present idealization provides good checks on the rotational symmetry of the results.

The ratio of outer to inner radius was set to 2.0. The following boundary conditions were imposed: On the two faces with x = constant

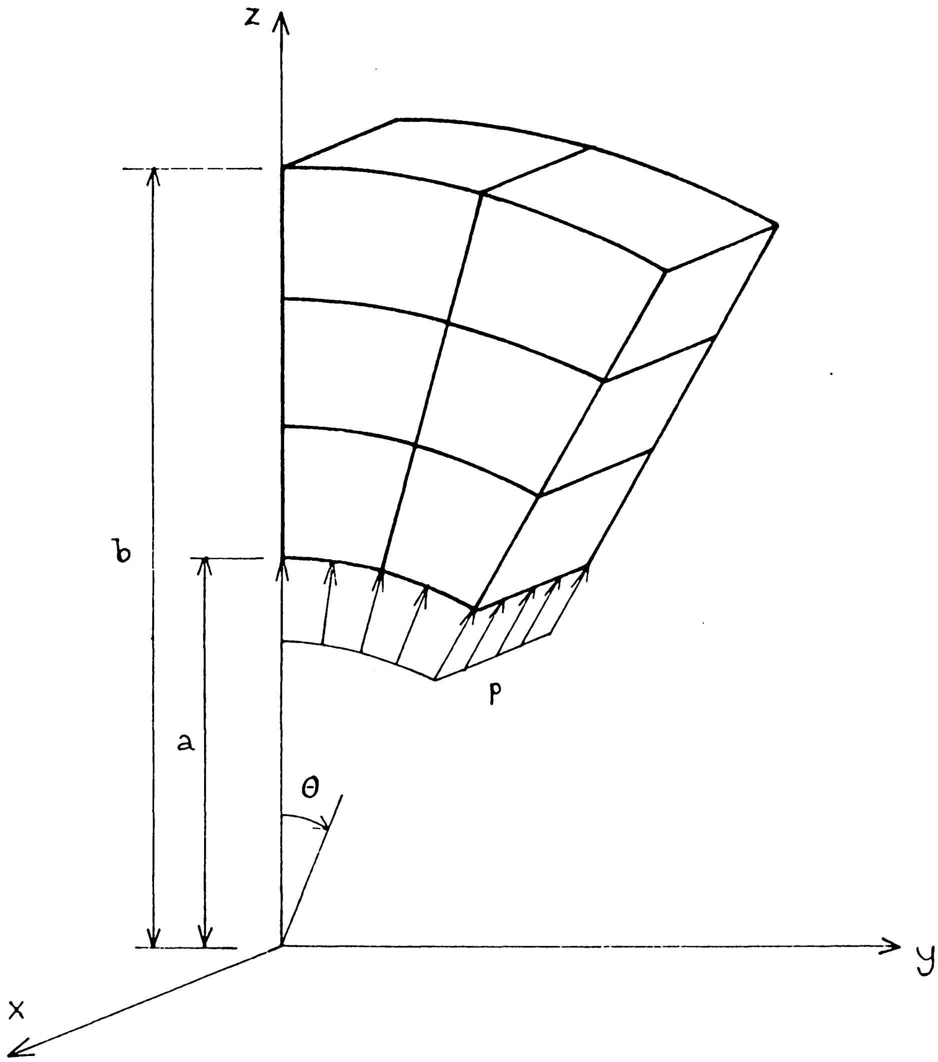


FIG. 9.1. Thick walled cylinder

no x-displacement was permitted. The nodal points at $\Theta = 0$ and $\Theta = 30°$ were on rollers that allowed movement only in the radial direction. On the remaining nodal points no restraints were imposed.

Nodal forces equivalent to the given pressure load were found with the aid of a short auxiliary program. The material constants chosen were the following:

$$E = 10^7 \text{ psi}$$

$$\nu = .33$$

$$\sigma_Y = 10^5 \text{ psi}$$

$$H' = 0$$

Ideal plasticity was assumed by setting the slope H' of the uniaxial stress - plastic strain curve to zero.

Following the linear solution, the elastic-plastic analysis was performed in eight load steps. The load increments 1 through 7 were 10% of the scaled linear load each and increment 8 was 3%. For the calculation of the average acceleration factor γ, the x-displacements in the middle surface were excluded, because they reflected only accidental changes inherent to the computer solution. The iteration within the first seven load increments was stopped after the largest absolute value of the residual forces was reduced to less than 5% of the original magnitude. The last iteration was discontinued after 24 cycles with the largest absolute value of the residual force component reduced to approximately 15% of its original size. The number of iterations per load step is given below.

Loadstep	p/σ_Y	No. of Iterations
1	.51	2
2	.56	2
3	.61	4
4	.65	5
5	.70	6
6	.75	9
7	.79	17
8	.81	24

The results are given at several stages of the loading process which is expressed in terms of the yield stress. Initial yield of the integration points closest to the inner surface of the cylinder was encountered at the load level of .47 σ_Y. Collapse was achieved at a load level of .81 σ_Y. The tangential stresses for different load steps are given in Fig. 9.2.

A comparison between the analytical solution given by Reckling [35] and the present finite-element analysis is made in Fig. 9.3. The load level at collapse for the analytic solution is computed as $p = .80\ \sigma_Y$ as compared to $p = .81\ \sigma_Y$ of the present solution.

B. Thin Plate with Transverse Loading

A square simply supported plate was analyzed as a second example. A first attempt with an idealization consisting of nine elements representing one quarter of the plate did not give good results. This was due to the fact that only one element was used through the thickness. Since

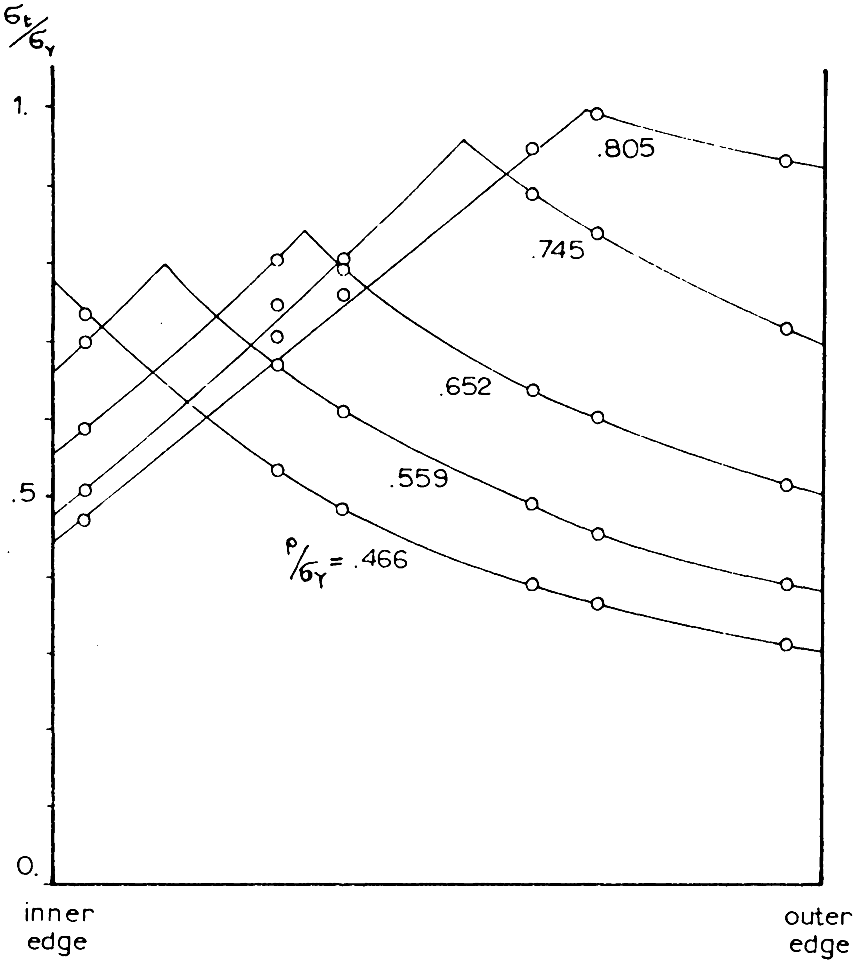


FIG. 9.2. σ_t/σ_Y for different loadings P/σ_Y

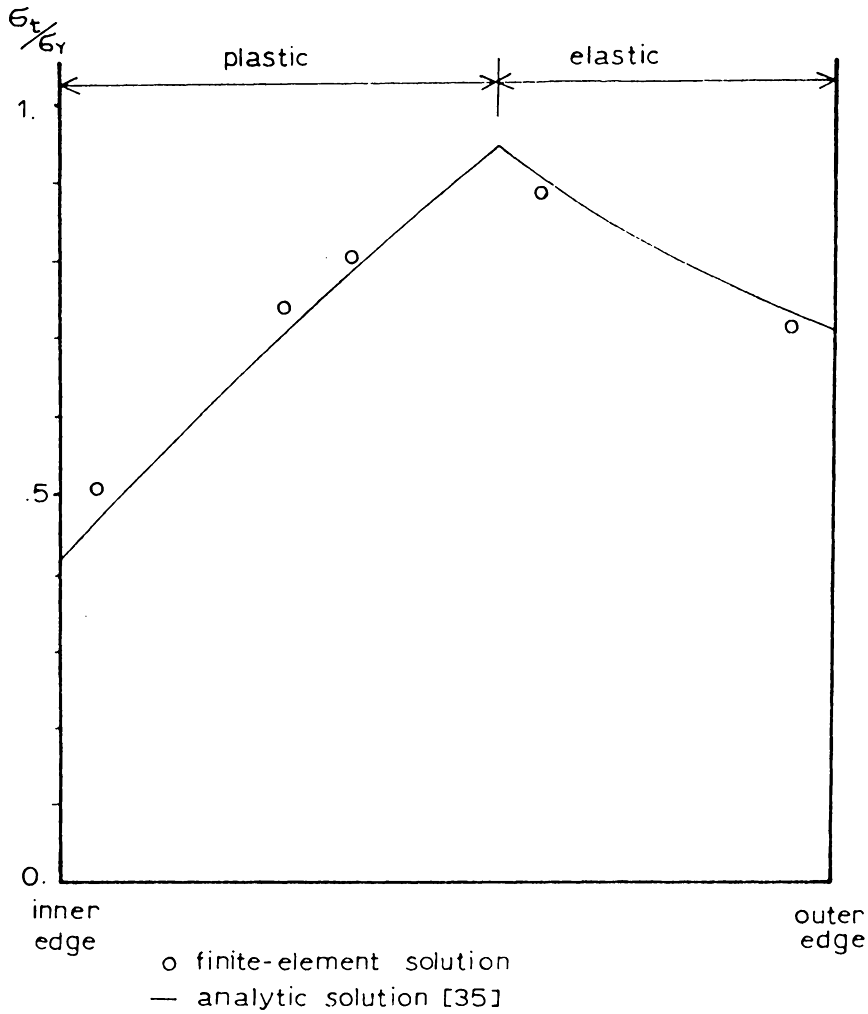


FIG. 9.3. Tangential stress σ_t for p/σ_Y =.745

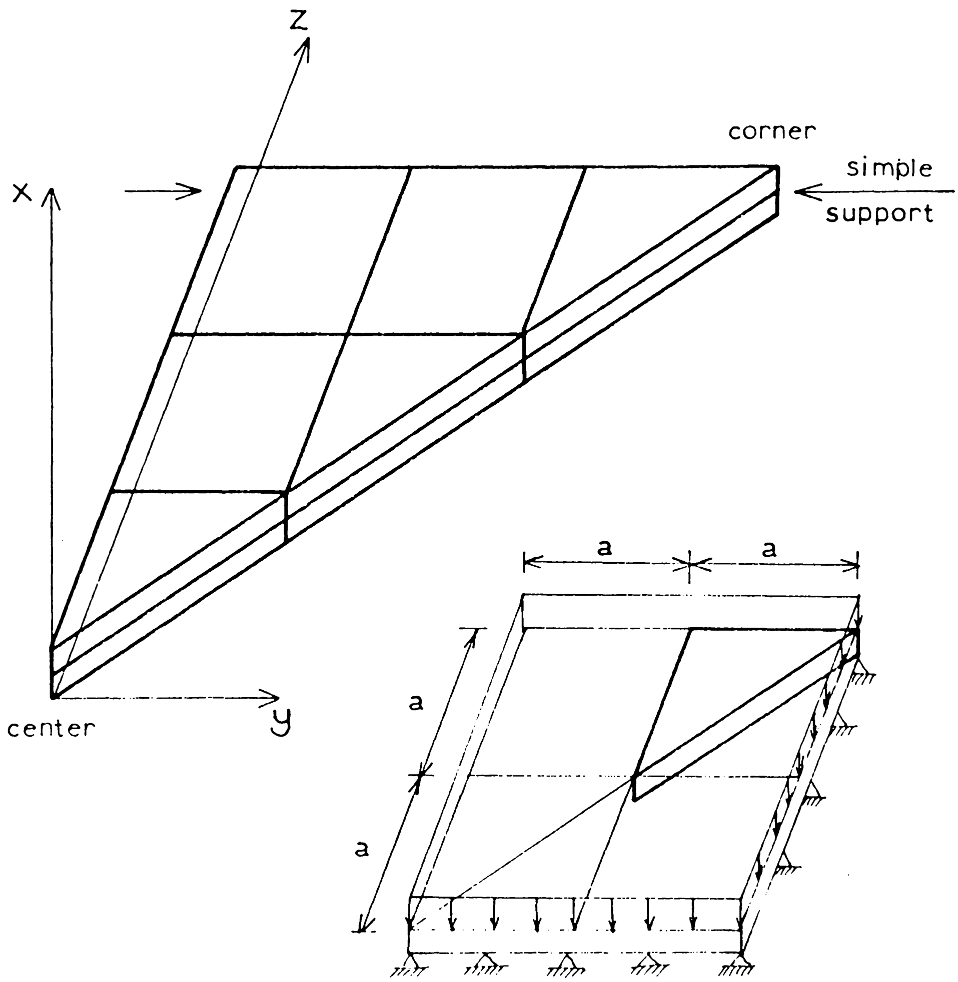


FIG. 9.4. Square plate idealization

such an element was yielding at the top and at the bottom simultaneously due to bending, all but the four integration points on the midsurface were in the plastic range. This meant that the load carrying capacity in bending for such an element appeared to be exhausted, even though almost 80% of the cross section was still elastic. Thus, progression of the yield surfaces from the top and bottom of the plate towards the middle surface could not be adequately modelled with only one layer of elements.

With a second model, the symmetry along the diagonal was taken into consideration as well. Only one eighth of the plate was analyzed. For reasons given above, two layers of elements were used. This idealization is depicted in Fig. 9.4. The distance "a" is 12 in. and the total thickness is .4 in. The corresponding aspect ratio of length to thickness is 60.

The nodal points common to the middle surface of the plate and the edge z = constant were simply supported against deflection in x-direction. In the plane of symmetry (y = 0) displacements in the y-direction were suppressed, and for the second plane of symmetry (y = z) no deflections perpendicular to that plane were allowed.

The loading was a uniformly distributed pressure p in x-direction. The material properties were the same as for the previous problem.

$$E = 10^7 \text{ psi}$$

$$\nu = .33$$

$$\sigma_Y = 10^5 \text{ psi}$$

$$H' = 0$$

The inelastic part of the solution was governed by ideal plasticity. The average acceleration factor was determined from the x-displacements of all nodal points except supports. A total of ten load steps of one tenth of the magnitude of the scaled linear solution were applied. During the last load step the iteration was concluded after 24 cycles with less than 15% of the original residual force component left. For all other increments the largest absolute value of the residual force components was reduced to less than 5% of the initial value. The number of iterations per load step is given below.

Loadstep	P	No. of Iterations
1	.57	2
2	.62	2
3	.68	4
4	.73	4
5	.78	5
6	.83	6
7	.88	7
8	.94	9
9	.99	15
10	1.04	24

The progression of the yield region on the top surface of the plate is shown for different load steps in Fig. 9.5. The loading p was normalized in the same fashion as in Armen, Pifko and Levine's paper [11], i.e.,

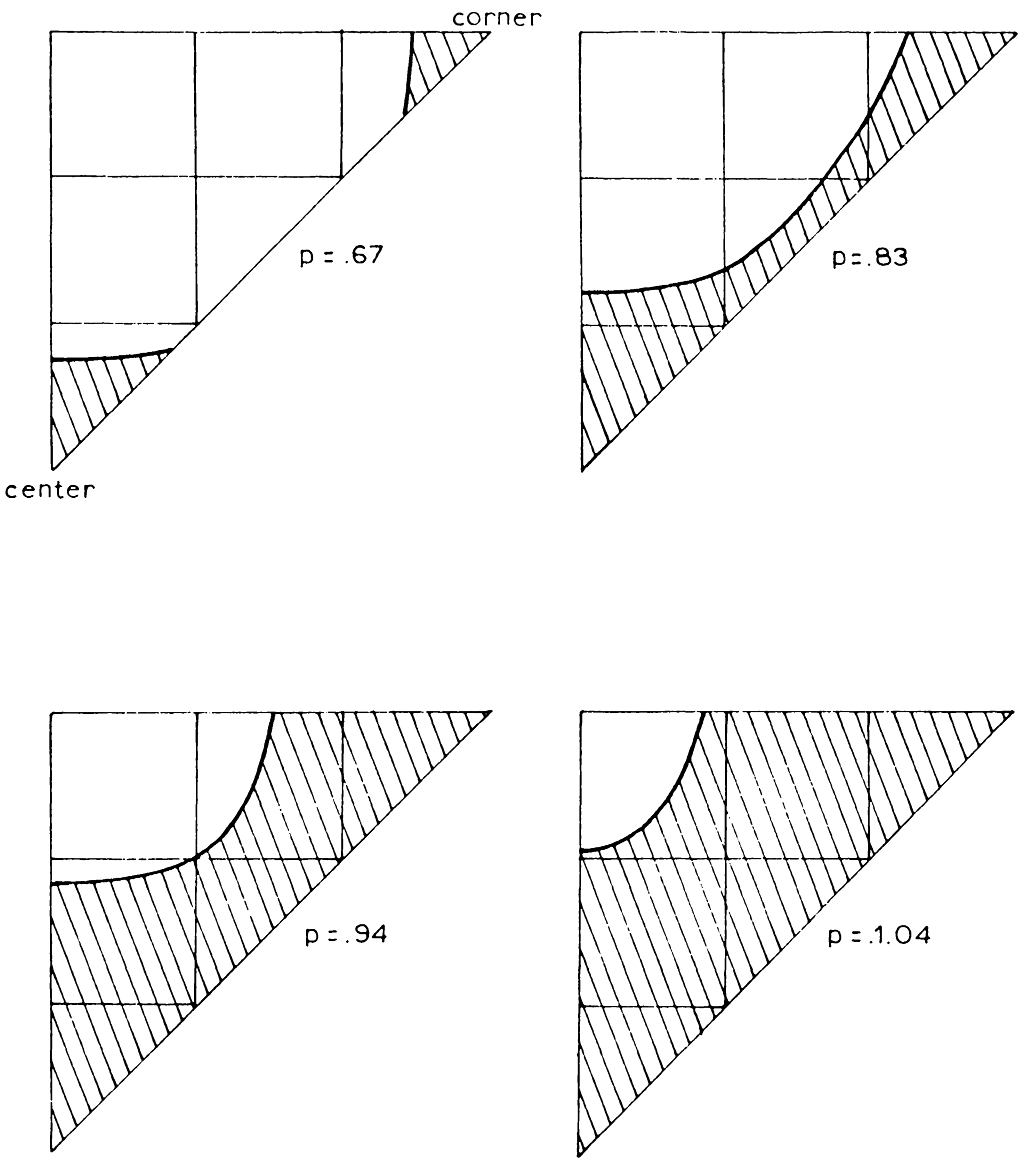


FIG. 9.5. Progression of yield region on plate surface

$$P = \frac{p\ a^2}{6M_o} \tag{9.1}$$

where

p = uniform load

a = half of width of plate

M_o = yield moment with section fully yielded.

The quantity M_o was computed from σ_Y and the thickness t of the plate as

$$M_o = \sigma_Y \frac{t^2}{4} \tag{9.2}$$

The factor P for which yielding was first encountered, at the corner and at the center of the plate almost simultaneously, was .52. Collapse was determined to take place at P = 1.04. A previously published finite-element investigation [11] determined the initial yield factor P as .50 and collapse was reached at 1.12.

Since with the present investigation a load level is sought for which convergence of the iterative process is still possible, a lower bound on the collapse load is actually obtained. The difference in the collapse load for the two solutions is 8%. The agreement between the two solutions is good, even though the two methods of analysis were totally different. Armen, Pifko and Levine [11] used a two-dimensional plate element with its characteristics determined by the location of the yield surface with respect to the plate thickness. In this application a general purpose three-dimensional element was used.

C. Fiber Composite Under Temperature Loading

The effect of a free edge on a fiber composite under a loading due to uniform change of temperature was examined. The fiber axes were perpendicular to the free surface. A hexagonal arrangement of the fibers with a volumetric ratio of matrix to fiber of 1:1 was chosen.

The finite-element idealization consisting of 24 elements is shown in Fig. 9.6. The thickness of the individual layers increases the further away they are from the free surface. At a distance from the free surface the stress distribution is assumed to be almost uniform. The analysis showed that the free edge effects dissipated within a few fiber diameters distance. On all three boundaries $y = 0$, $z = 0$ and $\Theta = 30°$, the nodal points are restrained against movement perpendicular to these planes. The top surface $x = 4.6$ is free, and the bottom surface is restrained so that it remains plane.

The fiber material is aluminum and the matrix is epoxy. The properties are listed below and the uniaxial stress-strain curves as well as the uniaxial stress-plastic strain curves are given in Fig. 9.7.

		Aluminum	Epoxy
E,	psi	10^7	$4.45\ 10^5$
ν,		.33	.35
Coeff. of therm. expansion,	deg C^{-1}	$2.4\ 10^{-5}$	$6.3\ 10^{-5}$
σ_Y,	psi	$1.5\ 10^4$	$.5\ 10^4$
H',	psi	$5.7\ 10^5$	$1.17\ 10^6$
$\varepsilon_e{}^p$,ult		.029	.006

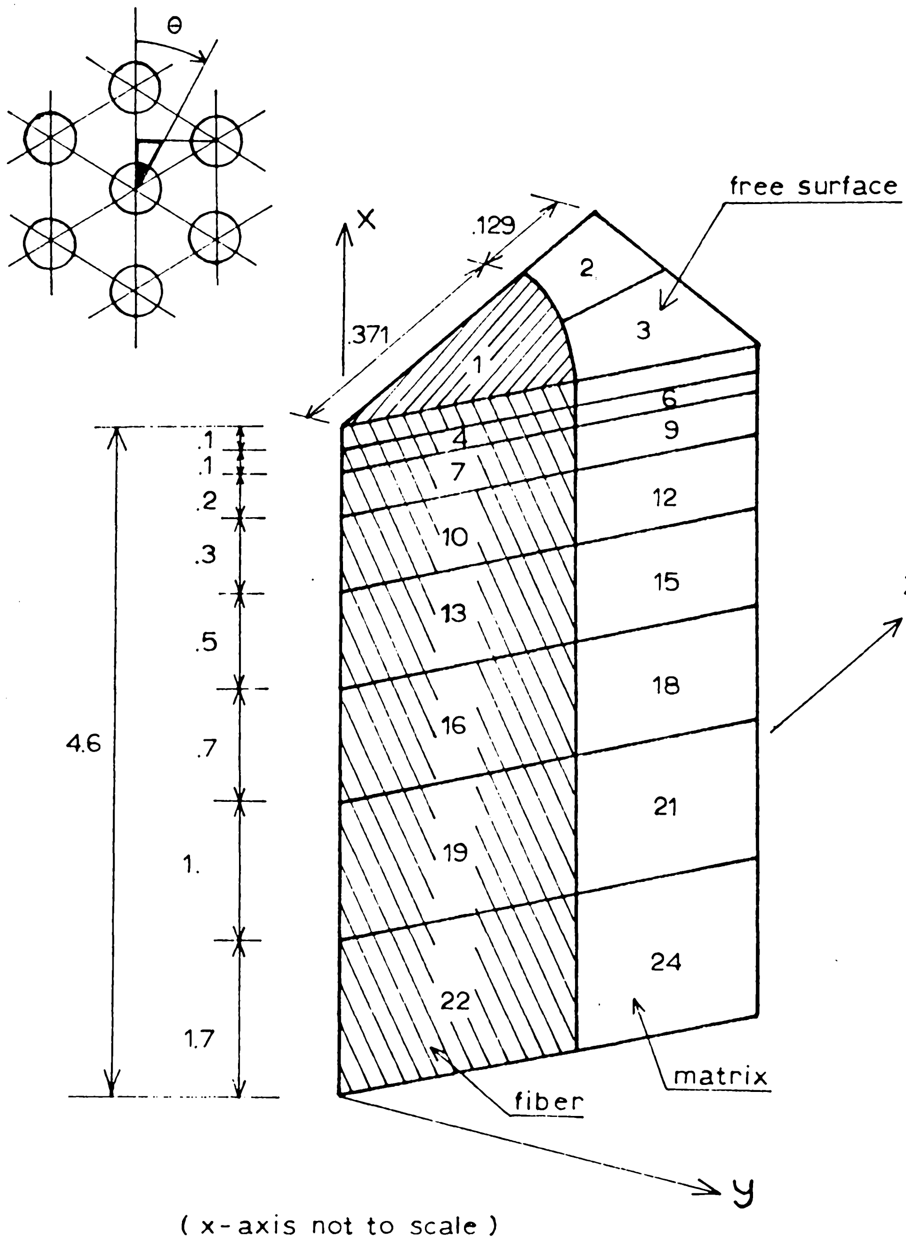
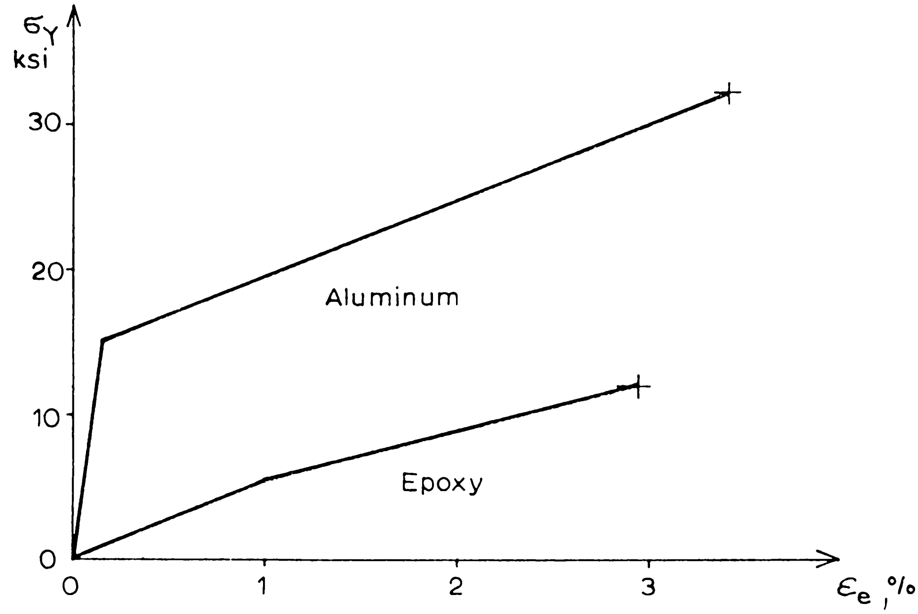


FIG. 9.6. Idealization of fiber composite

a) Uniaxial stress-total strain diagram



b) Uniaxial stress-plastic strain diagram

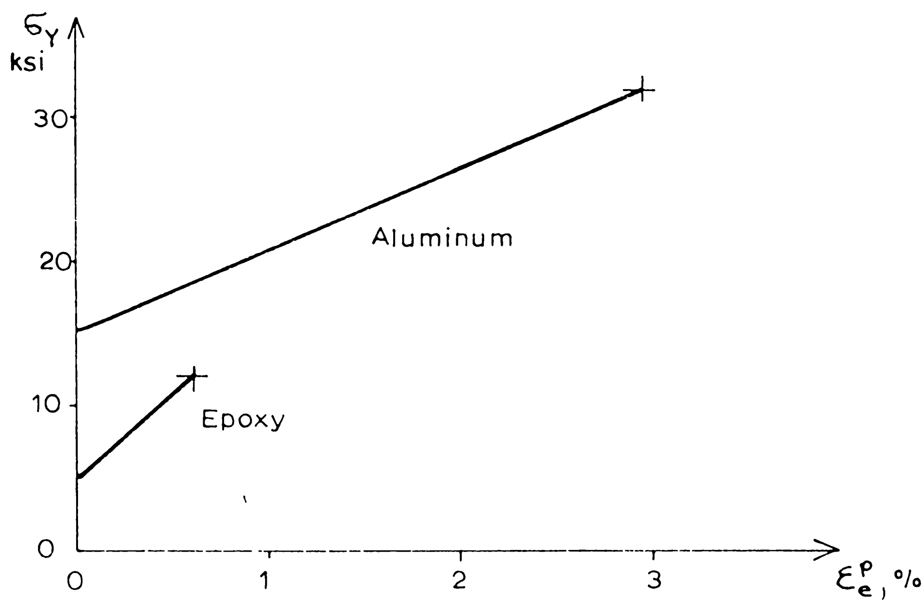


FIG. 9.7. Stress-strain diagrams

The elastic properties are adopted from an experimental investigation on an aluminum-epoxy composite, and the slopes of the uniaxial stress-plastic strain curves are approximated from a graph given in a finite-element investigation by Adams [20]. For this problem the elastic-plastic behavior for both materials is governed by linear strain hardening. The respective values of H' were kept constant throughout the inelastic part of the analysis.

Equivalent nodal forces due to the uniform temperature difference were computed for every element during the first linear analysis. A loading of this nature occurs during the curing cycle which is part of the manufacturing process of composites. The temperature difference T_Y that initiated yield in the matrix near the free surface was -47°C.

The inelastic part of the analysis was broken up into seven increments with each one adding 30% of the scaled linear loading to the previously established load level. At the end of the seventh load increment, the equivalent plastic strain $\varepsilon_e{}^p$ had reached the failure level for the matrix material in the vicinity of the free surface. The analysis was discontinued at this point because local failure of the epoxy near the free surface would have taken place. The existence and propagation of cracks cannot be modelled with the given method of analysis.

Since failure of the fiber composite specimen occurred only locally no significant increase in the number of iterations per load increment was observed. This, of course, is in contrast to the two preceding problems where the collapse of the total model was implied from the instability of the iterative solution. The number of iterations per load increment are as follows.

Load Increment	T/T_y	No. of Iterations
1	1.3	4
2	1.6	4
3	1.9	8
4	2.2	8
5	2.5	8
6	2.8	8
7	3.1	8

For every load step a short description of the results is given below. The location of the integration points for the two basic elements is illustrated in Fig. 9.8.

Linear solution: Temperature difference = -47°C

The σ_x stresses over the surface x = 0 are well balanced. Because of the fiber to matrix ratio of 1:1 these longitudinal stresses are of opposite sign but same magnitude.

Load increment 1: Temperature difference = -61°C

Integration points 6, 9 and 10 of both matrix elements next to the free surface are in the plastic range. Large shear stresses in the immediate neighborhood of the fiber contribute to this condition.

Load increment 2: Temperature difference = -75°C

Yielding continues for the integration points 6, 9 and 10 of elements 2 and 3. Point 14 of element 2 and point 13 of element 3 have also entered the plastic range.

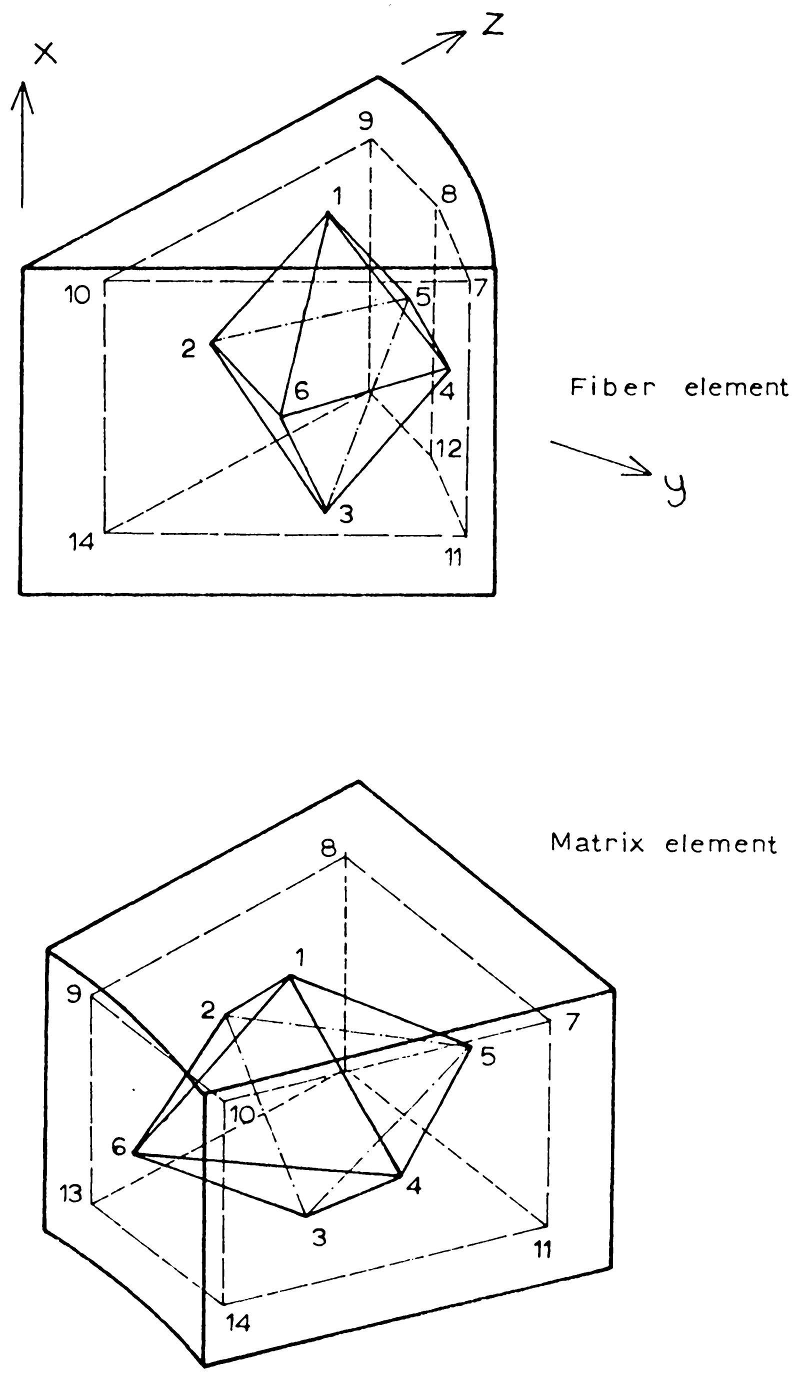


FIG. 9.8. Location of the integration points for two representative elements

Load increment 3: Temperature difference = -84°C

During this load increment large parts of the fiber have reached the elastic limit.

	Element	Integration Points Yielded
Fiber	1	1, 2, 5, 7-9, 12
	4	2, 5, 8, 11-13
	7	2, 5, 8, 11-13
	10	2-9, 11-14
	13, 16, 19, 22	1-14
Matrix	2	1-4, 6, 9-11, 13, 14
	3	1-4, 6, 9, 10, 12-14
	5	1, 7-10
	6	1, 7-11

The matrix elements show tensile normal stresses, σ_x, σ_y and σ_z of similar magnitude and relatively low shear stresses except in the neighborhood of the free surface. Their stress state is mainly hydrostatic and therefore not subject to yielding. Even though the epoxy is the weaker material, it remains in the elastic range. The normal stresses σ_y and σ_z of the fiber elements are tensile and σ_x is compressive. As a consequence, the deviatoric stresses for the fiber are high which in turn pushes the equivalent stress over the elastic limit for aluminum.

Load increment 4: Temperature difference = -104°C

The trend indicated by the previous load increments continues with the plastic region growing steadily.

	Element	Integration Points Yielded
Fiber	1	1, 2, 4-13
	4	2, 3, 5, 7-9, 11-13
	7	1-3, 5, 7-9, 11-13
	10, 13, 16, 19, 22	1-14
Matrix	2	1-6, 8-14
	3	1-4, 6, 9-14
	5	1, 2, 4-10, 13, 14
	6	1, 2, 4-14

Load increment 5: Temperature difference = -117°C

	Element	Integration Points Yielded
Fiber	1	1-13
	4	1-3, 5, 7-9, 11-13
	7	1-9, 11-14
	10, 13, 16, 19, 22	1-14
Matrix	2	1-14
	3	1-6, 9-14
	5, 6	1-14
	8	7, 8, 10
	9	1, 7-10

Load increment 6: Temperature difference = -131°C

	Element	Integration Points Yielded
Fiber	1	1-14
	4	1-9, 11-13
	7	1-9, 11-13
	10, 13, 16, 19, 22	1-14
Matrix	2	1-14
	3	1-6, 9-14
	5, 6	1-14
	8	1, 7-10
	9	1, 4, 6-10

Load increment 7: Temperature difference = -145°C

	Element	Integration Points Yielded
Fiber	1	1-14
	4	1-13
	7	1-9, 11-14
	10, 13, 16, 19, 22	1-14
Matrix	2, 3, 5, 6	1-14
	8	1, 2, 4, 5, 7-10
	9	1, 2, 4, 5, 7-10

The plastic strain $\varepsilon_e{}^p$ at integration point 9, element 2 has reached the value of .006 which is the maximum allowable plastic strain for epoxy as specified by the uniaxial stress-plastic strain curve. The distribution of the longitudinal shear stress τ_{xz} in the matrix along the interface is shown in Fig. 9.9 for both the elastic and the maximum loading. These shear stresses are largest where the distance between the fibers is the smallest, i.e., at $\theta = 0$.

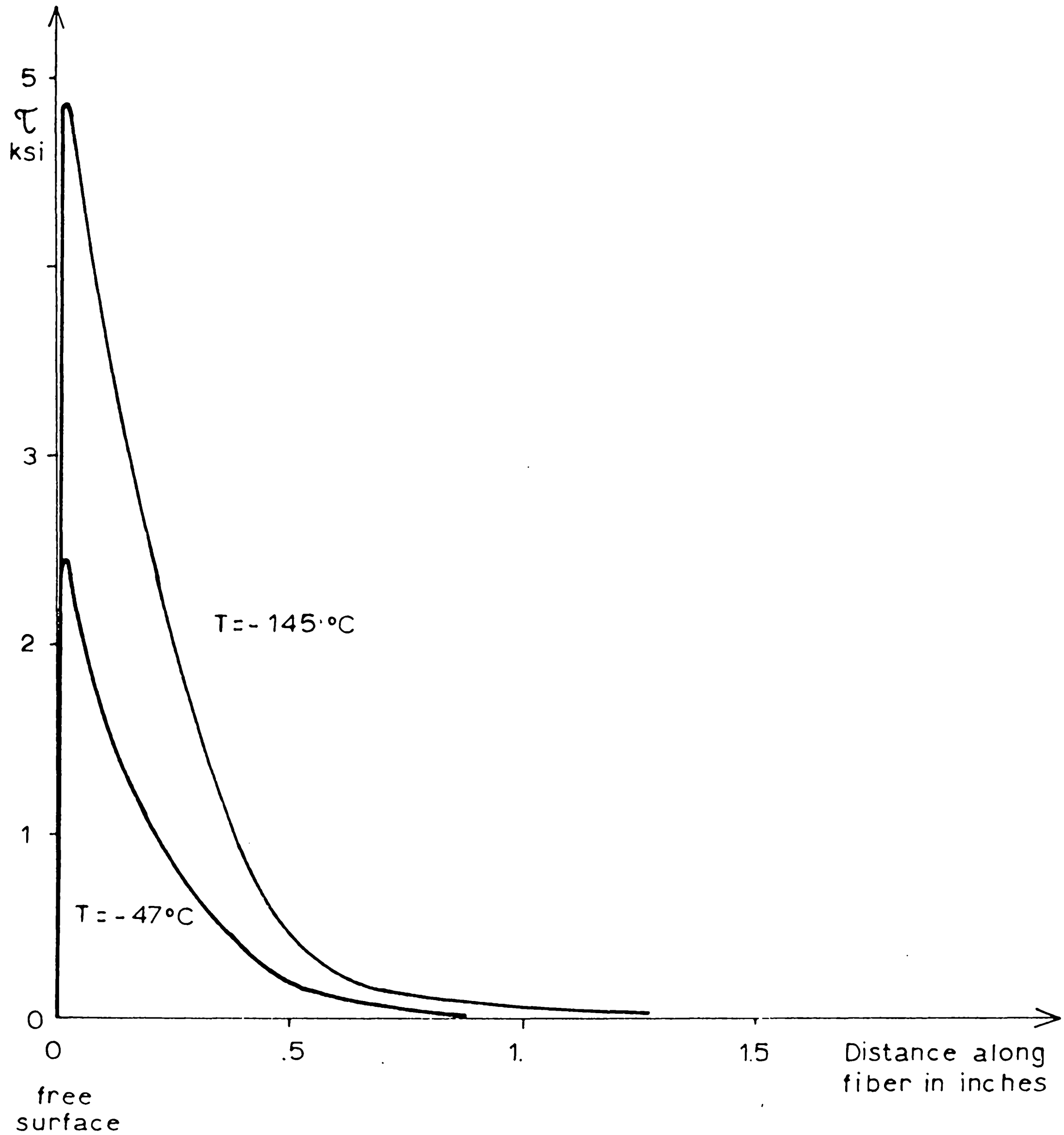


FIG. 9.9 Shear stress in matrix along fiber

X. CONCLUSIONS

In developing the described computer program and the subsequent application to three widely differing example problems, the following conclusions were drawn.

1. In creating the linear program designed for large problems (up to 3000 DOF), it was observed that the enforcement of boundary conditions as well as changing the coordinate system for selected boundary nodes was carried out with great advantage element-by-element on the individual stiffness matrices. Time consuming input and output operations encurred through block-by-block modification of the total stiffness matrix were avoided.
2. For the nonlinear part of the program, analyzing considerably smaller problems (approx. 600 DOF) because of the computational effort involved in the repeated linear solution, the uniform acceleration scheme proved to be an effective and useful feature. Convergence within a given load increment was faster without adding considerably to the individual solution time. This was largely due to the fact that this acceleration did not destroy equilibrium, and, therefore, the equivalent nodal forces were required to be evaluated only within the plastic region.
3. Observations on the average acceleration factor γ allowed conclusions to be drawn on the total force-displacement

response of the model. Usually for the first few load increments γ was close to 1.0, the response was almost exclusively elastic and convergence was achieved within a few iterations. If, on the other hand, displacements increased without additional loading, the value of γ approached 2.0. This indicated that the load carrying capacity was almost exhausted, a fact also reflected in a steep increase in the number of iterations per load increment.

4. The ideal plasticity problems, the thick cylinder and the square plate, showed good agreement with available analytical results in one case and with a special purpose finite-element analysis in the other. Collapse was indicated in both examples by the uniform acceleration factor γ approaching the value 2.0, and the number of iterations per increment rising dramatically.
5. The results of the third example with two different strain hardening materials indicated the total temperature difference for which local failure occurred. The location was given by the respective integration point for which the allowable equivalent plastic strain ε_e^p was reached first. Since, due to strain hardening, the load carrying capacity of the specimen was far from being exhausted when local failure occurred, no increase in number of iterations per load increment was observed.

BIBLIOGRAPHY

1. M. J. Turner, R. W. Clough, H. C. Martin and L. J. Topp, "Stiffness and Deflection Analysis of Complex Structures," J. Aeronaut. Sci., Vol. 25, No. 5, 1956, pp. 805-823.

2. A. A. Ilyushin, "Some Problems in the Theory of Plastic Deformation," RMB-12, transl. from Prikl. Math. Mech., 7, 1943, pp. 245-272, by Grad. Division Appl. Math., Brown University, 1946.

3. A. Mendelson and S. S. Manson, "Practical Solution of Plastic Deformation Problems in the Elastic-Plastic Range," NASA TR R28, 1959.

4. R. H. Gallagher, J. Padlog, and P. P. Bijlaard, "Stress Analysis of Heated Complex Shapes," J. of Am. Rocket Soc., Vol. 32, No. 5, 1962, pp. 700-707.

5. J. H. Argyris, "Elasto-plastic Matrix Displacement Analysis of Three-dimensional Continua," J. of Roy. Aer. Soc., Sept. 65, Vol. 69.

6. O. C. Zienkiewicz, S. Valliappan and I. P. King, "Elasto-plastic Solutions of Engineering Problems 'Initial Stress', Finite-Element Approach," Int. J. for Num. Meth. in Engr., Vol. 1, 1969, pp. 75-100.

7. G. Pope, "A Discrete Element Method for Analysis of Plane Elastic-plastic Stress Problems," Royal Aeronautical Establishment TR 65028, 1965.

8. J. L. Swedlow and W. H. Yang, "Stiffness Analysis of Elastic-plastic plates," Graduate Aeronautical Lab., California Institute of Technology SM 65-10, 1965.

9. P. V. Marcal and I. P. King, "Elastic-plastic Analysis of Two-dimensional Stress Systems by the Finite-Element Method," Int. J. Mech. 9, 3, 1967.

10. Y. Yamada, N. Yoshimura and T. Sakurai, "Plastic Stress-strain Matrix and its Application for the Solution of Elastic-plastic Problems by the Finite-element Method," Int. J. Mech. Sci, 1968, Vol. 10.

11. H. Armen, Jr., A. Pifko and H. S. Levine, "A Finite-element Method for the Plastic Bending Analysis of Structures," 2nd. Conf. on Matrix Meth. in Engr. Analysis, Wright-Patterson AFB, 1969, AFFDL-TR-68-150.

12. G. C. Nayak and O. C. Zienkiewicz, "Elasto-plastic Stress Analysis. A Generalization of Various Constitutive Relations including Strain Softening," Int. J. for Num. Meth. in Engr., Vol. 5, 1972, pp. 113-135.

13. G. Yagawa and Y. Ando, "Three-dimensional Finite-element Method of Thermoelastoplasticity with Creep Effect," 2nd, Int. Conf. on Struct. Mech. in Reactor Techn., Berlin, Sept. 1973.

14. P. Sharifi and D. N. Yates, "Nonlinear Thermo-elastic-plastic and Creep Analysis by the Finite-element Method," AIAA/ASME/SAE 14th Struct., Struct. Dyn. and Matl. Conf., Williamsburg, March 1973.

15. K. S. Havner, "On Convergence of Iterative Methods in Plastic Strain Analysis," Int. J. Solids Struct. 1968, Vol. 4, pp. 491-508.

16. A. Jennings, "Accelerating the Convergence of Matrix Iterative Processes," J. Inst. Maths. Applies, 1971, pp. 99-110.

17. E. F. Boyle and A. Jennings, "Accelerating the Convergence of Elastic-plastic Stress Analysis," Int. J. Num. Meth. Engr., Vol. 7, 1973, pp. 497-508.

18. G. C. Nayak and O. C. Zienkiewicz, "Note on the 'Alpha'-Constant Stiffness Method for the Analysis of Nonlinear Problems," Int. J. Num. Meth. Engr., Vol. 4, 1972, pp. 579-582.

19. Y. Yamamoto, "Rate of Convergence for the Iterative Approach in Elastic-plastic Analysis of Continua," Int. J. Num. Meth. Engr., Vol. 7, 1973, pp. 497-508.

20. D. F. Adams, "Inelastic Analysis of a Unidirectional Composite Subjected to Transverse Normal Loading," J. Comp. Matls., Vol. 4, 1970, pp. 310-328.

21. T. H. Lin, D. Salinas, Y. M. Ito, "Elastic-plastic Analysis of Unidirectional Composites," J. Comp. Matls., Vol. 6, 1972, pp. 48-60.

22. R. L. Foye, "Theoretical Post-yielding behavior of Composite Laminates Part I - Inelastic Micromechanics," J. Comp. Matls., Vol. 7, 1973, pp. 178-193.

23. R. L. Foye, "Theoretical Post-yielding Behavior of Composite Laminates Part II - Inelastic Macromechanics," J. Comp. Matls., Vol. 7, 1973, pp. 310-319.

24. D. R. J. Owen and J. F. Lyness, "Investigation of Bond Failure in Fibre Reinforced Materials by the Finite-element Method," Fibre Sci, Techn., Vol. 5, 1972, pp. 129-141.

25. N. K. Asamoah and W. G. Wood, "Shrinkage Stresses Near a Discontinuity in a Fibre Composite Material," J. of Strain Analy., Vol. 7, No. 9, 1972, pp. 54-60.

26. N. K. Asamoah and W. G. Wood, "Thermal Self-straining of Fibre Reinforced Materials," J. of Strain Analy., Vol. 5, No. 2, 1970, pp. 88-97.

27. A. E. Taylor, "Advanced Calculus," Blaisdell Publishing Company, 1955.

28. P. C. Hammer and A. H. Stroud, "Numerical Evaluation of Multiple Integrals II," Math. Tables and Aids of Computation, 12, 1958, pp. 272-280.

29. B. M. Irons, "Quadrature Rule for Brick Based Finite-elements," Int. J. Num. Meth. Engr., Vol. 3, 1971, pp. 293-294.

30. T. K. Hellen, "Effective Quadrature Rules for Quadratic Solid Isoparametric Finite Elements," Int. J. Num. Meth. Engr., Vol. 4, 1972, pp. 597-600.

31. D. C. Drucker, "A More Fundamental Approach to Plastic Stress-Strain Relations," 1st U. S. Congress of Applied Mechanics, ASME, New York, 1952, pp. 487-491.

32. L. Prandtl, "Spannungsverteilung in plastischen Koerpern," Proc. of 1st Int. Congress on Applied Mechanics, Delft, 1925, pp. 43-54.

33. E. Reuss, "Beruecksichtigung der elastischen Formaenderungen in der Plastizitaetstheorie," Z. Angew. Math. Mech., 20, 1930, pp. 226-274.

34. E. L. Wilson and W. P. Doherty, SESM Programs, University of California, Berkeley, September 1969.

35. K.-A. Reckling, "Plastizitaetstheorie und ihre Anwendung auf Festigkeitsprobleme," Springer Verlag, Berlin/Heidelberg/New York, 1967.

APPENDIX 1

$$x = \alpha_2 + \alpha_2\,\xi + \alpha_3\eta + \alpha_4\zeta + \alpha_5\xi\eta + \alpha_6\eta\zeta + \alpha_7\xi\zeta + \alpha_8\,\xi^2$$
$$+ \alpha_9\eta^2 + \alpha_{10}\zeta^2 + \alpha_{11}\xi\eta\zeta + \alpha_{12}\xi^2\eta + \alpha_{13}\xi^2\zeta + \alpha_{14}\xi\eta^2 + \alpha_{15}\eta^2\zeta$$
$$+ \alpha_{16}\xi\zeta^2 + \alpha_{17}\eta\zeta^2 + \alpha_{18}\xi^2\eta\zeta + \alpha_{19}\xi\eta^2\zeta + \alpha_{20}\xi\eta\zeta^2$$

The local coordinates of the integration points, the [CIP] matrix

0.795822426	0	0
0.0	-0.795822426	0.0
-0.795822426	0.0	0.0
0.0	0.795822426	0.0
0.0	0.0	0.795822426
0.0	0.0	-0.795822426
0.758786911	0.758786911	0.758786911
0.758786911	-0.758786911	0.758786911
0.758786911	-0.758786911	-0.758786911
0.758786911	0.758786911	-0.758786911
-0.758786911	0.758786911	0.758786911
-0.758786911	-0.758786911	0.758786911
-0.758786911	-0.758786911	-0.758786911
-0.758786911	0.758786911	-0.758786911

The coefficients of 8 x $[C]^{-1}$, the [ICI] Matrix

-2	2	-2	2	-2	2	-2	2	2	2	2	2	-2	2	-2	2	-2	2	-2	2
-1	2	-1	2	-1	2	-1	2	0	0	0	0	1	-2	-2	1	-2	-2	1	-2
-1	0	1	-2	1	0	-1	2	2	-2	-2	2	-1	0	1	-2	1	0	-1	2
-1	2	-1	0	1	-2	1	0	2	2	-2	-2	-1	2	-1	0	1	-2	1	0
0	0	0	-2	0	0	0	2	0	0	0	0	0	0	0	2	0	0	0	-2
0	0	0	0	0	0	0	0	2	-2	2	-2	0	0	0	0	0	0	0	0
0	2	0	0	0	-2	0	0	0	0	0	0	0	-2	0	0	0	2	0	0
1	0	1	0	1	0	1	0	-2	-2	-2	-2	1	0	1	0	1	0	1	0
1	-2	1	0	1	-2	1	0	0	0	0	0	1	-2	1	0	1	-2	1	0
1	0	1	-2	1	0	1	-2	0	0	0	0	1	0	1	-2	1	0	1	-2
1	0	-1	0	1	0	-1	0	0	0	0	0	-1	0	1	0	-1	0	1	0
1	0	-1	0	-1	0	1	0	-2	2	2	-2	1	0	-1	0	-1	0	1	0
1	0	1	0	-1	0	-1	0	-2	-2	2	2	1	0	1	0	-1	0	-1	0
1	-2	1	0	1	-2	1	0	0	0	0	0	-1	2	-1	0	-1	2	-1	0
1	-2	1	0	-1	2	-1	0	0	0	0	0	1	-2	1	0	-1	2	-1	0
1	0	1	-2	1	0	1	-2	0	0	0	0	-1	0	-1	2	-1	0	-1	2
1	0	-1	2	-1	0	1	-2	0	0	0	0	1	0	-1	2	-1	0	1	-2
1	0	-1	0	1	0	-1	0	-2	2	-2	2	1	0	-1	0	1	0	-1	0
1	-2	1	0	-1	2	-1	0	0	0	0	0	-1	2	-1	0	1	-2	1	0
1	0	-1	2	-1	0	1	-2	0	0	0	0	-1	0	1	-2	1	0	-1	2

APPENDIX 2

Partial Derivatives of {N} with respect to ξ, η, ζ

$$N^T = [1,\xi,\eta,\zeta,\xi\eta,\eta\zeta,\xi\zeta,\xi^2,\eta^2,\zeta^2,\xi\eta\zeta,\xi^2\eta,\xi^2\zeta,\xi\eta^2,\eta^2\zeta,\xi\zeta^2,\eta\zeta^2,$$
$$\xi^2\eta\zeta,\xi\eta^2\zeta,\xi\eta\zeta^2]\ [C]^{-1}$$

$$\left\{\frac{\partial N}{\partial \xi}\right\}^T = [0,1,0,0,\eta,0,\zeta,2\xi,0,0,\eta\zeta,2\xi\eta,2\xi\zeta,\eta^2,0,\zeta^2,0,2\xi\eta\zeta,\eta^2\zeta,\eta\zeta^2]\ [C]^{-1}$$

$$\left\{\frac{\partial N}{\partial \eta}\right\}^T = [0,0,1,0,\xi,\eta,0,0,2\eta,0,\xi\eta,\xi^2,0,2\xi\eta,2\eta\zeta,0,\zeta^2,\eta^2\zeta,2\xi\eta\zeta,\xi\zeta^2]\ [C]^{-1}$$

$$\left\{\frac{\partial N}{\partial \zeta}\right\}^T = [0,0,0,1,0,\eta,\zeta,0,0,2\zeta,\xi\eta,0,\xi^2,0,\eta^2,2\xi\eta,2\eta\zeta,\xi^2\eta,\xi\eta^2,2\xi\eta\zeta]\ [C]^{-1}$$

Components of the Jacobian Matrix (J)

$$\frac{\partial x}{\partial \xi} = \left\{\frac{\partial N}{\partial \xi}\right\}^T \{x_n\} \qquad \frac{\partial x}{\partial \eta} = \left\{\frac{\partial N}{\partial \eta}\right\}^T \{x_n\} \qquad \frac{\partial x}{\partial \zeta} = \left\{\frac{\partial N}{\partial \zeta}\right\}^T \{x_n\}$$

$$\frac{\partial y}{\partial \xi} = \left\{\frac{\partial N}{\partial \xi}\right\}^T \{y_n\} \qquad \frac{\partial y}{\partial \eta} = \left\{\frac{\partial N}{\partial \eta}\right\}^T \{y_n\} \qquad \frac{\partial y}{\partial \zeta} = \left\{\frac{\partial N}{\partial \zeta}\right\}^T \{y_n\}$$

$$\frac{\partial z}{\partial \xi} = \left\{\frac{\partial N}{\partial \xi}\right\}^T \{z_n\} \qquad \frac{\partial z}{\partial \eta} = \left\{\frac{\partial N}{\partial \eta}\right\}^T \{z_n\} \qquad \frac{\partial z}{\partial \zeta} = \frac{\partial N}{\partial \zeta}^T \{z_n\}$$

APPENDIX 3 Computer Listing

```
C       FEM  60 DOF  HEXAHEDRON
C       FINITE ELEMENT ANALYSIS USING A 60-DOF ISOPARAMETRIC
C       HEXAHEDRON WITH 3 DEGREES OF FREEDOM AT EACH CORNER- AND
C       MIDSIDE NODE
        IMPLICIT REAL * 8 (A-G,O-Z)
        INTEGER CODE
        COMMON /DAT/ CIP(14,3),X(1000),Y(1000),Z(1000),U(1000),V(1000),W(1
       1000),FE(60),EP(14,6),CODE(1000),IX(200,22),ICI(20,20),NUMNP,NUMEL,
       2NUMMAT
        COMMON /STI/ S(60,60),BEL(60),DEL(60),IDOF(60)
        COMMON /LAS/ E(8,8),D(6,6)
        COMMON /HAD/ HED(18),IPAGE,LINE
        COMMON /TOT/ A4(10000),B2(300),IBW,NUMEQ
        DIMENSION A1(10000), A2(10000)
        DIMENSION XX(1000), YY(1000), ZZ(1000)
        DIMENSION MAXB(300)
        EQUIVALENCE (U(1),XX(1)), (V(1),YY(1)), (W(1),ZZ(1))
        EQUIVALENCE (A4(1),A1(1))
        DEFINE FILE 1 (1000,8000,L,IJK)
        IPAGE=1
        READ (5,1) HED
1       FORMAT (18A4)
        CALL DOCUMT
        WRITE (9) HED
        WRITE (9) ((ICI(I,J),J=1,20),I=1,20)
        WRITE (9) ((CIP(I,J),J=1,3),I=1,14)
        WRITE (9) NUMEL,NUMNP,NUMMAT
        DO 2 I=1,NUMMAT
        WRITE (9) I,(E(J,I),J=1,5)
2       CONTINUE
        DO 5 I=1,NUMEL
        INEL=I
```

```
      MTYPE=IX(INEL,21)
      IF (INEL.EQ.1) GO TO 3
      INELL=INEL-1
      IF (IX(INEL,21).EQ.IX(INELL,21)) GO TO 4
3     CALL ELAS (MTYPE)
4     CONTINUE
      CALL ELSTIF (INEL)
      CALL STIFF (INEL)
5     CONTINUE
C
C     ARRANGE TOTAL STIFFNESS MATRIX IN BLOCKS LESS THAN 10000
C     STORAGE LOCATIONS. (SUBROUTINE RERITE)
C
      REWIND 1
      NUMEQ=10000/IBW
      NB=NUMEQ/3
      NUMEQ=NB*3
      NBLOCK=3*NUMNP/NUMEQ
      NUM=NBLOCK*NUMEQ
      NDOF=3*NUMNP
      IF (NUM.LT.NDOF) NBLOCK=NBLOCK+1
      NEQB=NUMEQ
      IF (NUMEQ.GT.NDOF) GO TO 6
      GO TO 7
6     NUMEQ=NDOF
      NBLOCK=1
7     CONTINUE
      CALL RERITE (A4,B2,NBLOCK,NDOF,IBW,NUMEQ)
      WRITE (6,8)
8     FORMAT (//,10X,13H ICI - MATRIX,//)
      WRITE (6,9) ((ICI(I,J),J=1,20),I=1,20)
9     FORMAT (5X,20I4)
```

```
      WRITE (6,10)
10    FORMAT (//,10X,13H CIP - MATRIX,//)
      WRITE (6,11) ((CIP(I,J),J=1,3),I=1,14)
11    FORMAT (5X,3D15.7)
      DO 12 I=1,NUMNP
      WRITE (9) I,CODE(I),X(I),Y(I),Z(I),U(I),V(I),W(I)
12    CONTINUE
      DO 13 I=1,NUMEL
      WRITE (9) I,(IX(I,J),J=1,22)
13    CONTINUE
      LL=1
      NEQB=NUMEQ
      MBAND=IBW
      NSB=(MBAND+LL)*NEQB
      N1=2
      N2=3
      N3=4
      N4=8
      N5=4
      CALL USOL (A1,A2,MAXB,NEQB,MBAND,LL,NBLOCK,NSB,N1,N2,N3,N4,N5)
C
C     READ DISPLACEMENTS FROM DISK (4) AND PRINT THEM OUT
C
      REWIND 4
      NNP=NEQB/3
      DO 15 I=1,NBLOCK
      READ (4) ((XX(J),YY(J),ZZ(J)),J=1,NNP)
      IF (I.EQ.NBLOCK) GO TO 15
      NA=(NBLOCK-I)*NNP+1
      DO 14 K=1,NNP
      KK=NA+K-1
      XX(KK)=XX(K)
```

```
      YY(KK)=YY(K)
14    ZZ(KK)=ZZ(K)
15    CONTINUE
      CALL TITLE
      WRITE (6,16)
16    FORMAT (1H0,10X,4HNODE,7X,1HX,9X,1HY,9X,1HZ,13X,7HX-DISPL,13X,7HY-
     1DISPL,13X,7HZ-DISPL)
      DO 20 I=1,NUMNP
      LINE=LINE+1
      WRITE (6,17) I,XX(I),YY(I),ZZ(I)
17    FORMAT (11X,I3,31X,3D20.8)
      IF (LINE.GE.50) GO TO 18
      GO TO 19
18    CALL TITLE
      WRITE (6,16)
19    CONTINUE
20    CONTINUE
      DO 21 I=1,NUMNP
      IF (CODE(I).LE.7) GO TO 21
C
C     FOR ROTATED BOUNDARY CONDITIONS, EXPRESS DISPLACEMENTS
C     IN GLOBAL COORDINATES
C
      VV=V(I)
      WW=W(I)
      TANALF=1.7320508D0/2.0D0
      ALFA=DATAN(TANALF)
      AR22=DCOS(ALFA)
      AR33=AR22
      AR23=-DSIN(ALFA)
      AR32=-AR23
      V(I)=VV*AR22-WW*AR23
```

```
      W(I)=-VV*AR32+WW*AR33
21    CONTINUE
C
C     CALL STRESS ROUTINE AND FIND LARGEST ABSOLUTE VALUE OF EQUIVALENT
C     STRESS. EVALUATE SCALE FACTOR
C
      DO 24 I=1,NUMEL
      INEL=I
      IP=1
      CALL ELASTR (INEL,IP,SF)
      IF (INEL.EQ.1) GO TO 22
      IF (SF.LT.OSF) GO TO 22
      GO TO 23
22    OSF=SF
      IPO=IP
      INELO=INEL
23    CONTINUE
24    CONTINUE
      CALL TITLE
      WRITE (6,25) OSF,INELO,IPO
25    FORMAT (////,10X,20HTHE SCALE FACTOR IS ,D15.5,//,10X,53HTHE LOCAT
     1ION WHERE YIELDING IS FIRST ENCOUNTERED IS  ,/,54X,9HELEMENT  ,5X,
     2I5,/,44X,19HINTEGRATION POINT  ,5X,I5)
C
C     TRANSFER REDUCED EQUATIONS FROM SCRATCH AREA TO DIRECT ACCESS
C      STORAGE.
C
      REWIND 2
      DO 26 I=1,NBLOCK
      READ (3) (A1(J),J=1,NSB),(MAXB(J),J=1,NEQB)
      WRITE (2) (A1(J),J=1,NSB),(MAXB(J),J=1,NEQB)
26    CONTINUE
```

```
WRITE (9) NEQB,IBW,NBLOCK
WRITE (9) OSF,INELO,IPO
RETURN
END
```

```
      SUBROUTINE TITLE
      COMMON /HAD/ HED(18),IPAGE,LINE
      WRITE (6,1) IPAGE
1     FORMAT (1H1,3X,39H60-DOF GENERAL HEXAHEDRON ELEMENT, HATT,50X,4HPA
     1GE,I3)
      WRITE (6,2) HED
2     FORMAT (1H0,10X,18A4)
      IPAGE=IPAGE+1
      LINE=0
      RETURN
      END
```

```
      SUBROUTINE DOCUMT
C
C     PROBLEM DESCRIPTION IS READ IN AND PRINTED OUT
C
      IMPLICIT REAL * 8 (A-G,O-Z)
      INTEGER CODE
      COMMON /DAT/ CIP(14,3),X(1000),Y(1000),Z(1000),U(1000),V(1000),W(1
     1000),FE(60),EP(14,6),CODE(1000),IX(200,22),ICI(20,20),NUMNP,NUMEL,
     2NUMMAT
      COMMON /LAS/ E(8,8),D(6,6)
      COMMON /HAD/ HED(18),IPAGE,LINE
      COMMON /TOT/ A4(10000),B2(300),IBW,NUMEQ
      DIMENSION A(3,833), BE(3)
      EQUIVALENCE (A4(1),A(1)), (B2(1),BE(1))
C
C     READ INVERTED (C)-MATRIX RELATING THE NATURAL COORDINATES
C     TO THE CARTESIAN SYSTEM AND (CIP)-MATRIX GIVING THE LOCAL
C     COORDINATES OF THE 14 INTEGRATION POINTS
C
      READ (5,1) ((ICI(I,J),J=1,20),I=1,20)
      READ (5,2) ((CIP(I,J),J=1,3),I=1,14)
C
C     GENERAL PROBLEM INFORMATION
C
      READ (5,3) NUMNP,NUMEL,NUMMAT
    1 FORMAT (20I4)
    2 FORMAT (4D20.9)
    3 FORMAT (3I5)
      CALL TITLE
      WRITE (6,4) NUMNP,NUMEL,NUMMAT
    4 FORMAT (1H0,/,10X,28HNUMBER OF NODAL POINTS......,I3,/,10X,28HNUMB
     1ER OF ELEMENTS...........,I3,/,10X,28HNUMBER OF DIFF. MATERIALS...,
```

```
      2I3)
       DO 6 I=1,NUMMAT
       READ (5,5) MTYPE,(E(J,MTYPE),J=1,5)
       WRITE (6,7) MTYPE,(E(J,MTYPE),J=1,5)
5      FORMAT (I5,5D10.5)
6      CONTINUE
7      FORMAT (1H0,/,10X,9HMAT'L NO.,8X,1HE,15X,3HXNU,6X,22HCOEFF. OF THE
      1RMAL EXP.,4X,11HTEMP. DIFF.,6X,19HUNIAX. YIELD STRESS,/,(14X,I3,2D
      216.5,3D20.5))
       CALL TITLE
       WRITE (6,8)
8      FORMAT (1H0,3X,11HNODAL POINT,3X,4HTYPE,3X,8HX-COORD.,3X,8HY-COORD
      1.,3X,8HZ-COORD.,3X,15HX-(LOAD/DISPL.),3X,15HY-(LOAD/DISPL.),3X,15H
      2Z-(LOAD/DISPL.))
C
C      READ AND WRITE NODAL POINT DATA
C
       DO 10 I=1,NUMNP
       READ (5,12) K,CODE(K),X(K),Y(K),Z(K),U(K),V(K),W(K)
       WRITE (6,11) K,CODE(K),X(K),Y(K),Z(K),U(K),V(K),W(K)
       IF (LINE.LT.49) GO TO 9
       CALL TITLE
       WRITE (6,8)
       GO TO 10
9      LINE=LINE+1
10     CONTINUE
11     FORMAT (8X,I3,8X,I2,F10.3,2F11.3,1X,3D18.5)
12     FORMAT (2I5,6D10.3)
       CALL TITLE
       WRITE (6,13)
13     FORMAT (1H0,2X,3HEL.,121X,5HMAT'L,/,3X,3HNO.,119H          A         B          C
      1         D         E         F         G         H         I         J         K         L         M          N
```

```
     2       O       P       Q       R       S       T,3X,3HNO.)
C
C     READ AND WRITE ELEMENT DATA
C
      DO 16 J=1,NUMEL
      READ (5,17) M,(IX(M,I),I=1,22)
      IF (LINE.LT.50) GO TO 14
      CALL TITLE
      WRITE (6,13)
      GO TO 15
14    LINE=LINE+1
15    CONTINUE
      WRITE (6,18) M,(IX(M,I),I=1,21)
16    CONTINUE
17    FORMAT (16I5)
18    FORMAT (1X,I4,20I6,I4)
      NDIFF=0
      DO 21 I=1,NUMEL
      DO 21 J=1,19
      K=J+1
      DO 21 L=K,20
      IDIFF=IABS(IX(I,J)-IX(I,L))
      IF (IDIFF-NDIFF) 20,20,19
19    NDIFF=IDIFF
20    CONTINUE
21    CONTINUE
      IBW=3*NDIFF+3
      WRITE (6,22) IBW
22    FORMAT (//,5X,29HHALFBANDWIDTH OF PROBLEM IS =,I5,//)
C
C     CLEAR DISC SPACE TO BE OCCUPIED BY TOTAL STIFFNESS MATRIX
C
```

```
      DO 23 I=1,3
      BE(I)=0.0D0
      DO 23 J=1,IBW
23    A(I,J)=0.0D0
      DO 24 I=1,NUMNP
      IJK=I
      WRITE (1'IJK) ((A(J,K),K=1,IBW),BE(J),J=1,3)
24    CONTINUE
      DO 25 I=1,NUMNP
      IF (CODE(I).LE.7) GO TO 25
      GO TO 26
25    CONTINUE
26    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ELSTIF (INEL)
C
C     CALCULATION OF ELEMENT STIFFNESS MATRIX USING A 14 POINT
C     INTEGRATION RULE
C
      IMPLICIT REAL * 8 (A-G,O-Z)
      INTEGER CODE,COD
      COMMON /DAT/ CIP(14,3),X(1000),Y(1000),Z(1000),U(1000),V(1000),W(1
     1000),FE(60),EP(14,6),CODE(1000),IX(200,22),ICI(20,20),NUMNP,NUMEL,
     2NUMMAT
      COMMON /LAS/ E(8,8),D(6,6)
      COMMON /STI/ S(60,60),BEL(60),DEL(60),IDOF(60)
      DIMENSION DNXSI(20), DNETA(20), DNZET(20), DETJ(14), BB(6,3), BA(6
     1,60), B(84,60), BTD(6), XX(20), YY(20), ZZ(20)
      DIMENSION SM(60,60)
      DIMENSION SDI(60)
      EQUIVALENCE (B(1),SM(1))
      DO 1 I=1,20
      IDOF(3*I-2)=3*IX(INEL,I)-2
      IDOF(3*I-1)=IDOF(3*I-2)+1
      IDOF(3*I)=IDOF(3*I-2)+2
1     CONTINUE
      IF (INEL.EQ.1) GO TO 2
      IF (IX(INEL,22).EQ.1.OR.IX(INEL,22).EQ.3) GO TO 46
2     CONTINUE
C
C     NODAL COORDINATES OF ELEMENT
C
      DO 3 I=1,20
      II=IX(INEL,I)
      XX(I)=X(II)
      YY(I)=Y(II)
```

```
      ZZ(I)=Z(II)
3     CONTINUE
C
C     WEIGHT FACTORS FOR INTEGRATION
C
      BC=0.88642659300
      CC=0.33518OC55DO
C
C     ESTABLISH (B)-MATRIX FOR EVERY INTEGRATION POINT
C
      DO 11 N=1,14
      AJ11=0.0D0
      AJ12=0.0D0
      AJ13=0.0D0
      AJ21=0.0D0
      AJ22=0.0D0
      AJ23=0.0D0
      AJ31=0.0D0
      AJ32=0.0D0
      AJ33=0.0D0
      DO 4 I=1,20
      XSI=CIP(N,1)
      ETA=CIP(N,2)
      ZET=CIP(N,3)
C
C     PARTIAL DERIVATIVES OF N'S WITH RESPECT TO XSI,ETA AND ZETA
C
      DNXSI(I)=ICI(2,I)+ETA*ICI(5,I)+ZET*ICI(7,I)+2.0D0*XSI*ICI(8,I)+ETA
     1*ZET*ICI(11,I)+2.0D0*XSI*ETA*ICI(12,I)+2.0D0*XSI*ZET*ICI(13,I)+ETA
     2*ETA*ICI(14,I)+ZET*ZET*ICI(16,I)+2.0D0*XSI*ETA*ZET*ICI(18,I)+ETA*E
     3TA*ZET*ICI(19,I)+ETA*ZET*ZET*ICI(20,I)
      DNETA(I)=ICI(3,I)+XSI*ICI(5,I)+ZET*ICI(6,I)+2.0D0*ETA*ICI(9,I)+XSI
```

```
     1*ZET*ICI(11,I)+XSI*XSI*ICI(12,I)+2.0D0*XSI*ETA*ICI(14,I)+2.0D0*ETA
     2*ZET*ICI(15,I)+ZET*ZET*ICI(17,I)+XSI*XSI*ZET*ICI(18,I)+2.0D0*XSI*E
     3TA*ZET*ICI(19,I)+XSI*ZET*ZET*ICI(20,I)
      DNZET(I)=ICI(4,I)+ETA*ICI(6,I)+XSI*ICI(7,I)+2.0D0*ZET*ICI(10,I)+XS
     1I*ETA*ICI(11,I)+XSI*XSI*ICI(13,I)+ETA*ETA*ICI(15,I)+2.0D0*XSI*ZET*
     2ICI(16,I)+2.0D0*ETA*ZET*ICI(17,I)+XSI*ETA*XSI*ICI(18,I)+XSI*ETA*ET
     3A*ICI(19,I)+2.0D0*XSI*ETA*ZET*ICI(20,I)
4     CONTINUE
C
C     CALCULATE JACOBIAN
C
      DO 5 I=1,20
      AJ11=AJ11+DNXSI(I)*XX(I)
      AJ12=AJ12+DNXSI(I)*YY(I)
      AJ13=AJ13+DNXSI(I)*ZZ(I)
      AJ21=AJ21+DNETA(I)*XX(I)
      AJ22=AJ22+DNETA(I)*YY(I)
      AJ23=AJ23+DNETA(I)*ZZ(I)
      AJ31=AJ31+DNZET(I)*XX(I)
      AJ32=AJ32+DNZET(I)*YY(I)
      AJ33=AJ33+DNZET(I)*ZZ(I)
5     CONTINUE
      DJ=AJ11*AJ22*AJ33+AJ12*AJ23*AJ31+AJ13*AJ21*AJ32-AJ12*AJ21*AJ33-AJ1
     11*AJ23*AJ32-AJ13*AJ22*AJ31
C
C     INVERT JACOBIAN
C
      RJ11=1.0D0/DJ*(AJ22*AJ33-AJ23*AJ32)
      RJ22=1.0D0/DJ*(AJ11*AJ33-AJ13*AJ31)
      RJ33=1.0D0/DJ*(AJ11*AJ22-AJ12*AJ21)
      RJ21=1.0D0/DJ*(-AJ21*AJ33+AJ23*AJ31)
      RJ31=1.0D0/DJ*(AJ21*AJ32-AJ22*AJ31)
```

```
      RJ32=1.0D0/DJ*(-AJ11*AJ32+AJ12*AJ31)
      RJ12=1.0D0/DJ*(-AJ12*AJ33+AJ13*AJ32)
      RJ13=1.0D0/DJ*(AJ12*AJ23-AJ13*AJ22)
      RJ23=1.0D0/DJ*(-AJ11*AJ23+AJ13*AJ21)
      DETJ(N)=DJ/512.0D0
      DO 6 I=1,6
      DO 6 J=1,3
    6 BB(I,J)=0.0D0
C
C     PARTIAL DERIVATIVES OF N'S WITH RESPECT TO X,Y AND Z
C
      DO 8 I=1,20
      DNX=RJ11*DNXSI(I)+RJ12*DNETA(I)+RJ13*DNZET(I)
      DNY=RJ21*DNXSI(I)+RJ22*DNETA(I)+RJ23*DNZET(I)
      DNZ=RJ31*DNXSI(I)+RJ32*DNETA(I)+RJ33*DNZET(I)
      BB(1,1)=DNX
      BB(2,2)=DNY
      BB(3,3)=DNZ
      BB(4,1)=DNY
      BB(4,2)=DNX
      BB(5,2)=DNZ
      BB(5,3)=DNY
      BB(6,1)=DNZ
      BB(6,3)=DNX
      J=3*I-2
      K=3*I-1
      L=3*I
      DO 7 M=1,6
      BA(M,J)=BB(M,1)
      BA(M,K)=BB(M,2)
      BA(M,L)=BB(M,3)
    7 CONTINUE
```

```
8       CONTINUE
C
C       PLACE (BA)-MATRIX (6,60) IN (B)-MATRIX (84,60)
C
        DO 10 I=1,6
        K=6*(N-1)+I
        DO 9 J=1,60
        B(K,J)=BA(I,J)
9       CONTINUE
10      CONTINUE
11      CONTINUE
        IF (IX(INEL,22).LT.2) GO TO 34
C
C       EQUIVALENT THERMAL LOADS FOR UNIFORM TEMPERATURE DIFFERENCE.
C
        DO 12 I=1,60
12      FE(I)=0.0D0
        DO 13 I=1,14
        DO 13 J=1,6
13      EP(I,J)=0.0D0
        DO 14 I=1,14
        DO 14 J=1,3
        MTYPE=IX(INEL,21)
14      EP(I,J)=E(3,MTYPE)*E(4,MTYPE)
        DO 19 I=1,60
        DO 18 N=1,14
        SS=0.0D0
        DO 15 J=1,6
15      BTD(J)=0.0D0
        DO 16 J=1,6
        DO 16 K=1,6
        LL=6*(N-1)+K
```

```
16    BTD(J)=BTD(J)+B(LL,I)*D(K,J)
      DO 17 J=1,6
17    SS=SS+BTD(J)*EP(N,J)
      IF (N.LE.6) WF=BC
      IF (N.GT.6) WF=CC
      FE(I)=FE(I)+SS*WF*DETJ(N)
18    C3NTINUE
19    CJNTINUE
20    CONTINUE
      WRITE (6,21) INEL,(FE(I),I=1,60)
21    FORMAT (5X,41HEQUIVALENT THERMAL LOADS FOR ELEMENT NR =,I5,/,(5X,3
     1D15.6))
      DO 33 I=1,20
      N=IX(INEL,I)
      NN=3*I-2
      IF (CODE(N).GT.7) GO TO 22
      GO TO 23
22    CJD=CCDE(N)-8
      NP=IX(INEL,I)
C
C     FOR ROTATED BOUNDARY, EXPRESS THERMAL LOADS IN LOCAL COORDINATES
C
      TANALF=1.7320508D0/2.0D0
      ALFA=DATAN(TANALF)
      A11=1.0D0
      A12=0.0D0
      A13=0.0D0
      A21=0.0D0
      A31=0.0D0
      A22=DCOS(ALFA)
      A33=A22
      A23=DSIN(ALFA)
```

```
      A32=-A23
      BELL=A11*FE(NN)+A21*FE(NN+1)+A31*FE(NN+2)
      BELLL=A12*FE(NN)+A22*FE(NN+1)+A32*FE(NN+2)
      BELIV=A13*FE(NN)+A23*FE(NN+1)+A33*FE(NN+2)
      FE(NN)=BELL
      FE(NN+1)=BELLL
      FE(NN+2)=BELIV
23    CONTINUE
      IF (CODE(N).LE.7) COD=CODE(N)
      IF (COD.EQ.0) GO TO 31
      GO TO (24,25,26,27,28,29,30), COD
24    V(N)=V(N)+FE(NN+1)
      W(N)=W(N)+FE(NN+2)
      GO TO 32
25    W(N)=W(N)+FE(NN+2)
      GO TO 32
26    V(N)=V(N)+FE(NN+1)
      GO TO 32
27    U(N)=U(N)+FE(NN)
      W(N)=W(N)+FE(NN+2)
      GO TO 32
28    U(N)=U(N)+FE(NN)
      V(N)=V(N)+FE(NN+1)
      GO TO 32
29    U(N)=U(N)+FE(NN)
      GO TO 32
30    GO TO 32
31    U(N)=U(N)+FE(NN)
      V(N)=V(N)+FE(NN+1)
      W(N)=W(N)+FE(NN+2)
32    CONTINUE
33    CONTINUE
```

```
34    CONTINUE
      IF (IX(INEL,22).EQ.3) GO TO 48
C
C     CALCULATE UPPER TRIANGLE OF ELEMENT STIFFNESS MATRIX
C
      DO 35 I=1,60
      DO 35 J=1,60
35    S(I,J)=0.0D0
      DO 40 I=1,60
      DO 40 J=I,60
      DO 39 N=1,14
      SS=0.0D0
      DO 36 IJ=1,6
36    BTD(IJ)=0.0D0
      DO 37 K=1,6
      DO 37 L=1,6
      LL=6*(N-1)+L
      BTD(K)=BTD(K)+B(LL,I)*D(L,K)
37    CONTINUE
      DO 38 K=1,6
      KK=6*(N-1)+K
      SS=SS+BTD(K)*B(KK,J)
38    CONTINUE
      IF (N.LE.6) WF=BC
      IF (N.GT.6) WF=CC
      S(I,J)=S(I,J)+SS*WF*DETJ(N)
39    CONTINUE
40    CONTINUE
C
C     FILL IN LOWER TRIANGLE OF ELEMENT STIFFNESS MATRIX
C
      DO 41 IK=2,60
```

```
      LK=IK-1
      DO 41 JK=1,LK
41    S(IK,JK)=S(JK,IK)
      INELN=INEL+1
      IF (IX(INELN,22).EQ.0.OR.IX(INELN,22).EQ.2) GO TO 43
      DO 42 I=1,60
      DO 42 J=1,60
42    SM(I,J)=S(I,J)
43    CONTINUE
      DO 45 I=1,20
      NP=IX(INEL,I)
C
C     FOR ROTATED BOUNDARY CALL ROTBC TO MODIFY ELEMENT STIFFNESS
C     MATRIX ACCORDINGLY
C
      IF (CODE(NP).LE.7) GO TO 45
      CALL ROTBC (INEL)
      INELN=INEL+1
      IF (IX(INELN,22).EQ.0.OR.IX(INELN,22).EQ.2) GO TO 49
      DO 44 J=1,60
      DO 44 K=1,60
44    SM(J,K)=S(J,K)
      GO TO 49
45    CONTINUE
      GO TO 49
46    CONTINUE
      DO 47 I=1,60
      DO 47 J=1,60
47    S(I,J)=SM(I,J)
      IF (IX(INEL,22).EQ.3) GO TO 20
48    CONTINUE
49    CONTINUE
```

```
C
C     IDOF-VECTOR INDICATES THE GLOBAL DEGREE OF FREEDOM NUMBERS
C     OF THE ELEMENT
C
C     ACCORDING TO BOUNDARY CODES GIVE MINUS SIGN TO EACH DOF
C     WHERE A DISPLACEMENT IS SPECIFIED.
C     BEL-VECTOR CONTAINS NODAL FORCES
C     DEL-VECTOR CONTAINS NODAL DISPLACEMENTS
C
      MOD=0
      DO 59 INODE=1,20
      NP=IX(INEL,INODE)
      IF (CODE(NP).LE.7) COD=CODE(NP)
      IF (CODE(NP).GT.7) COD=CODE(NP)-8
      IF (COD.EQ.0) GO TO 58
      GO TO (50,51,52,53,54,55,56), COD
50    IDOF(3*INODE-2)=-IDOF(3*INODE-2)
      GO TO 57
51    IDOF(3*INODE-2)=-IDOF(3*INODE-2)
      IDOF(3*INODE-1)=-IDOF(3*INODE-1)
      GO TO 57
52    IDOF(3*INODE-2)=-IDOF(3*INODE-2)
      IDOF(3*INODE)=-IDOF(3*INODE)
      GO TO 57
53    IDOF(3*INODE-1)=-IDOF(3*INODE-1)
      GO TO 57
54    IDOF(3*INODE)=-IDOF(3*INODE)
      GO TO 57
55    IDOF(3*INODE-1)=-IDOF(3*INODE-1)
      IDOF(3*INODE)=-IDOF(3*INODE)
      GO TO 57
56    IDOF(3*INODE-2)=-IDOF(3*INODE-2)
```

```
      IDOF(3*INODE-1)=-IDOF(3*INODE-1)
      IDOF(3*INODE)=-IDOF(3*INODE)
57    MOD=1
58    CONTINUE
59    CONTINUE
      IF (MOD.EQ.0) GO TO 62
      WRITE (6,60) INEL
60    FORMAT (/,5X,29HMODIFIED DOF OF ELEMENT NR = ,I5)
      WRITE (6,61) (IDOF(I),I=1,60)
61    FORMAT (/,(3X,20I6))
62    CONTINUE
      DO 69 I=1,20
      INP=IX(INEL,I)
      II=3*I-2
      BEL(II)=0.0D0
      DEL(II)=0.0D0
      IF (IDOF(II).LT.0) GO TO 63
      BEL(II)=U(INP)
      U(INP)=0.0D0
      GO TO 64
63    DEL(II)=U(INP)
64    III=3*I-1
      BEL(III)=0.0D0
      DEL(III)=0.0D0
      IF (IDOF(III).LT.0) GO TO 65
      BEL(III)=V(INP)
      V(INP)=0.0D0
      GO TO 66
65    DEL(III)=V(INP)
66    IV=3*I
      BEL(IV)=0.0D0
      DEL(IV)=0.0D0
```

```
      IF (IDOF(IV).LT.0) GO TO 67
      BEL(IV)=W(INP)
      W(INP)=0.0D0
      GO TO 68
67    DEL(IV)=W(INP)
68    CONTINUE
69    CONTINUE
C
C     CALL SUBROUTINE MODIFY, IF NECESSARY
C
      IF (MOD.EQ.1) CALL MODIFY
      WRITE (6,70) INEL
70    FORMAT (/,3X,31HDIAGONAL COEFF. , ELEMENT NR = ,I5)
      DO 71 I=1,60
      SDI(I)=S(I,I)
71    CONTINUE
      WRITE (6,72) (SDI(I),I=1,60)
72    FORMAT (//,(2X,10D12.4))
      RETURN
      END
```

```
      SUBROUTINE MODIFY
C
C     MODIFICATION OF ELEMENT STIFFNESS MATRIX ACCORDING TO
C     PRESCRIBED BOUNDARY CONDITIONS
C
      IMPLICIT REAL * 8 (A-G,O-Z)
      COMMON /STI/ S(60,60),BEL(60),DEL(60),IDOF(60)
      DO 4 I=1,60
      IF (IDOF(I).LT.0) GO TO 3
      DO 2 J=1,60
      IF (IDOF(J).GT.0) GO TO 1
      BEL(I)=BEL(I)-S(I,J)*DEL(J)
1     CONTINUE
2     CONTINUE
3     CONTINUE
4     CONTINUE
      DO 7 I=1,60
      IF (IDOF(I).GT.0) GO TO 6
      DO 5 J=1,60
      S(J,I)=0.0D0
      S(I,J)=0.0D0
5     CONTINUE
      BEL(I)=DEL(I)
      IDOF(I)=-IDOF(I)
6     CONTINUE
7     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ELAS (MTYPE)
C
C     CALCULATION OF ELASTICITY MATRIX (D)
C
      IMPLICIT REAL * 8 (A-G,O-Z)
      COMMON /LAS/ E(8,8),D(6,6)
      EE=E(1,MTYPE)
      XNU=E(2,MTYPE)
      DO 1 I=1,6
      DO 1 J=1,6
1     D(I,J)=0.0D0
      CONST=EE*(1.0D0-XNU)/((1.0D0+XNU)*(1.0D0-2.0D0*XNU))
      D(1,1)=CONST
      D(1,2)=CONST*XNU/(1.0D0-XNU)
      D(1,3)=D(1,2)
      D(2,2)=CONST
      D(2,3)=D(1,2)
      D(3,3)=CONST
      D(4,4)=CONST*(1.0D0-2.0D0*XNU)/(2.0D0*(1.0D0-XNU))
      D(5,5)=D(4,4)
      D(6,6)=D(4,4)
      D(2,1)=D(1,2)
      D(3,1)=D(1,3)
      D(3,2)=D(2,3)
      RETURN
      END
```

```
      SUBROUTINE STIFF (INEL)
C
C     COMPILATION OF OVERALL STIFFNESS MATRIX IN RECTANGULAR FORM
C     DIAGONAL ELEMENTS IN FIRST COLUMN
C
      IMPLICIT REAL * 8 (A-G,O-Z)
      INTEGER CODE
      COMMON /DAT/ CIP(14,3),X(1000),Y(1000),Z(1000),U(1000),V(1000),W(1
     1000),FE(60),EP(14,6),CODE(1000),IX(200,22),ICI(20,20),NUMNP,NUMEL,
     2NUMMAT
      COMMON /STI/ S(60,60),BEL(60),DEL(60),IDOF(60)
      COMMON /TOT/ A4(10000),B2(300),IBW,NUMEQ
      DIMENSION IIDOF(500), IIIDOF(500), IVDOF(500)
      DIMENSION A(3,833), BE(3)
      EQUIVALENCE (A4(1),A(1)), (B2(1),BE(1))
C
C     TRANSFER ELEMENT STIFFNESS MATRIX NODE BY NODE
C
      DO 10 I=1,20
      DO 1 J=1,IBW
      IIDOF(J)=0
      IIIDOF(J)=0
1     IVDOF(J)=0
      KOUNT=1
      JK=3*I-2
      JL=3*I-1
      JM=3*I
      DO 3 J=1,20
      JJ=3*J-2
      DO 2 K=1,3
      IIDOF(KOUNT)=IDOF(JJ)-IDOF(JK)+K
      IIIDOF(KOUNT)=IDOF(JJ)-IDOF(JL)+K
```

```
      IVDOF(KOUNT)=IDOF(JJ)-IDOF(JM)+K
      KOUNT=KCUNT+1
    2 CONTINUE
    3 CONTINUE
      IJK=IX(INEL,I)
C
C     READ THE THREE LINES CORRESPONDING TO THE NODAL POINT BEING
C     PROCESSED AND ADD ELEMENT STIFFNESS COEFFICIENTS TO TOTAL
C     STIFFNESS COEFFICIENTS
C
      READ (1'IJK) ((A(J,K),K=1,IBW),BE(J),J=1,3)
      IS=3*I-2
      BE(1)=BE(1)+BEL(IS)
      DO 5 M=1,60
      IF (IIDOF(M).LT.1) GO TO 4
      MM=IIDOF(M)
      A(1,MM)=A(1,MM)+S(IS,M)
    4 CONTINUE
    5 CONTINUE
      IS=3*I-1
      BE(2)=BE(2)+BEL(IS)
      DO 7 M=1,60
      IF (IIIDOF(M).LT.1) GO TO 6
      MM=IIIDOF(M)
      A(2,MM)=A(2,MM)+S(IS,M)
    6 CONTINUE
    7 CONTINUE
      IS=3*I
      BE(3)=BE(3)+BEL(IS)
      DO 9 M=1,60
      IF (IVDOF(M).LT.1) GO TO 8
      MM=IVDOF(M)
```

```
      A(3,MM)=A(3,MM)+S(IS,M)
 8    CONTINUE
 9    CONTINUE
      IJK=IX(INEL,I)
C
C     WRITE BACK ON DISC
C
      WRITE (1'IJK) ((A(J,K),K=1,IBW),BE(J),J=1,3)
10    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RERITE (A,BE,NBLOCK,NDOF,IBW,NUMEQ)
C
C     REWRITE TOTAL STIFFNESS MATRIX INTO FORM COMPATIBLE WITH
C     SUBROUTINE USOL. FORM BLOCKS OF APPROXIMATELY 10000 STORAGE
C     LOCATIONS. NUMBER OF EQUATIONS PER BLOCK IS DEPENDENT ON
C     BANDWIDTH.
C
      IMPLICIT REAL * 8 (A-G,O-Z)
      INTEGER CODE
      COMMON /DAT/ CIP(14,3),X(1000),Y(1000),Z(1000),U(1000),V(1000),W(1
     1000),FE(60),EP(14,6),CODE(1000),IX(200,22),ICI(20,20),NUMNP,NUMEL,
     2NUMMAT
      DIMENSION A(NUMEQ,IBW), BE(NUMEQ)
      REWIND 1
      NBL=NBLOCK-1
      NREQ=NDOF-NBL*NUMEQ
      IF (NBLOCK.EQ.1) NREQ=NUMEQ
      NEQB=NUMEQ
      IF (NBL.EQ.0) GO TO 6
      DO 5 K=1,NBL
      NIJK=NUMEQ/3
      DO 1 L=1,NIJK
      NA=3*L-2
      NB=3*L
      IJK=(K-1)*NIJK+L
    1 READ (1'IJK) ((A(I,J),J=1,IBW),BE(I),I=NA,NB)
      DO 3 J=1,NUMEQ
      IF (A(J,1).LT.0.1D-14) GO TO 2
      GO TO 3
    2 A(J,1)=1.0D0
      KDOF=(K-1)*NEQB+J
      INOD=(KDOF-1)/3+1
```

```
      NNDOF=3*INOD
      IF (NNDOF-2.EQ.KDOF) BELL=U(INOD)
      IF (NNDOF-1.EQ.KDOF) BELL=V(INOD)
      IF (NNDOF.EQ.KDOF) BELL=W(INOD)
      BE(J)=BELL
3     CONTINUE
      DO 4 J=1,NIJK
      IJL=(K-1)*NIJK+J
      U(IJL)=BE(3*J-2)
      V(IJL)=BE(3*J-1)
4     W(IJL)=BE(3*J)
5     WRITE (2) ((A(I,J),I=1,NUMEQ),J=1,IBW),(BE(I),I=1,NUMEQ)
6     CONTINUE
C
C     FOR EQUATIONS WHERE DISPLACEMENTS ARE PRESCRIBED, DIAGONAL
C     ELEMENT HAS TO BE SET TO 1.0 AND THE DISPLACEMENT BE(J) HAS TO BE
C     RESET TO ITS ORIGINAL VALUE BECAUSE OF POSSIBLE ACCUMULATION IN
C     SUBROUTINE STIFF
C
C     REARRANGE LAST BLOCK AND FILL REMAINING EQUATIONS WITH ZEROS
      NIJK=NREQ/3
      DO 7 L=1,NIJK
      NA=3*L-2
      NB=3*L
      IJK=NBL*NUMEQ/3+L
7     READ (1'IJK) ((A(I,J),J=1,IBW),BE(I),I=NA,NB)
      DO 9 J=1,NREQ
      IF (A(J,1).LT.0.1D-14) GO TO 8
      GO TO 9
8     A(J,1)=1.0D0
      KDOF=NBL*NEQB+J
      INOD=(KDOF-1)/3+1
```

```
      NNDOF=3*INOD
      IF (NNDOF-2.EQ.KDOF) BELL=U(INOD)
      IF (NNDOF-1.EQ.KDOF) BELL=V(INOD)
      IF (NNDOF.EQ.KDOF) BELL=W(INOD)
      BE(J)=BELL
9     CONTINUE
      IF (NREQ.EQ.NUMEQ) GO TO 11
      II=NREQ+1
      DO 10 I=II,NUMEQ
      BE(I)=0.0D0
      DO 10 J=1,IBW
10    A(I,J)=0.0D0
11    CONTINUE
      DO 12 L=1,NIJK
      IJL=NBL*NUMEQ/3+L
      U(IJL)=BE(3*L-2)
      V(IJL)=BE(3*L-1)
12    W(IJL)=BE(3*L)
      WRITE (2) ((A(I,J),I=1,NUMEQ),J=1,IBW),(BE(I),I=1,NUMEQ)
      RETURN
      END
```

```
      SUBROUTINE ROTBC (INEL)
C
C     IF DISPLACEMENT BOUNDARY CONDITIONS PRESCRIBED ARE NOT
C     IN THE GLOBAL COORDINATE SYSTEM, THE AFFECTED STIFFNESS
C     COEFFICIENTS HAVE TO BE MODIFIED.
C
      IMPLICIT REAL * 8 (A-G,O-Z)
      INTEGER CODE
      COMMON /DAT/ CIP(14,3),X(1000),Y(1000),Z(1000),U(1000),V(1000),W(1
     1000),FE(60),EP(14,6),CODE(1000),IX(200,22),ICI(20,20),NUMNP,NUMEL,
     2NUMMAT
      COMMON /STI/ S(60,60),BEL(60),DEL(60),IDOF(60)
      DIMENSION ST(60,60)
      WRITE (6,1) INEL
1     FORMAT (//,10X,33HSUBROUTINE ROTBS ENTERED, EL.NO.=,I5)
      DO 2 I=1,20
      NP=IX(INEL,I)
      IF (CODE(NP).LE.7) GO TO 2
      IDOF(3*I-2)=-IDOF(3*I-2)
      IDOF(3*I-1)=-IDOF(3*I-1)
      IDOF(3*I)=-IDOF(3*I)
2     CONTINUE
      DO 6 I=1,60,3
      IF (IDOF(I).GT.0) GO TO 6
      II=(I+2)/3
      NP=IX(INEL,II)
C
C     ALFA IS ANGLE ASSOCIATED WITH NODAL POINT IX(INEL,II)
C
      TANALF=1.7320508D0/2.0D0
      ALFA=DATAN(TANALF)
      A11=1.0D0
```

```
      A12=0.0D0
      A13=0.0D0
      A21=0.0D0
      A31=0.0D0
      A22=DCOS(ALFA)
      A33=A22
      A23=DSIN(ALFA)
      A32=-A23
      DO 5 J=1,60,3
      IF (J.EQ.I) GO TO 4
      IF (IDOF(J).LT.0) GO TO 3
      ST(I,J)=A11*S(I,J)+A21*S(I+1,J)+A31*S(I+2,J)
      ST(I,J+1)=A11*S(I,J+1)+A21*S(I+1,J+1)+A31*S(I+2,J+1)
      ST(I,J+2)=A11*S(I,J+2)+A21*S(I+1,J+2)+A31*S(I+2,J+2)
      ST(I+1,J)=A12*S(I,J)+A22*S(I+1,J)+A32*S(I+2,J)
      ST(I+1,J+1)=A12*S(I,J+1)+A22*S(I+1,J+1)+A32*S(I+2,J+1)
      ST(I+1,J+2)=A12*S(I,J+2)+A22*S(I+1,J+2)+A32*S(I+2,J+2)
      ST(I+2,J)=A13*S(I,J)+A23*S(I+1,J)+A33*S(I+2,J)
      ST(I+2,J+1)=A13*S(I,J+1)+A23*S(I+1,J+1)+A33*S(I+2,J+1)
      ST(I+2,J+2)=A13*S(I,J+2)+A23*S(I+1,J+2)+A33*S(I+2,J+2)
      ST(J,I)=ST(I,J)
      ST(J+1,I)=ST(I,J+1)
      ST(J+2,I)=ST(I,J+2)
      ST(J,I+1)=ST(I+1,J)
      ST(J+1,I+1)=ST(I+1,J+1)
      ST(J+2,I+1)=ST(I+1,J+2)
      ST(J,I+2)=ST(I+2,J)
      ST(J+1,I+2)=ST(I+2,J+1)
      ST(J+2,I+2)=ST(I+2,J+2)
      GO TO 5
 3    IF (J.LT.I) GO TO 5
      JJ=(J+2)/3
```

```
      NP=IX(INEL,JJ)
C
C     BETA IS ANGLE ASSOCIATED WITH NODAL POINT IX(INEL,JJ)
C
      TANBET=1.7320508D0/2.0D0
      BETA=DATAN(TANBET)
      B11=1.0D0
      B12=0.0D0
      B13=0.0D0
      B21=0.0D0
      B31=0.0D0
      B22=DCOS(BETA)
      B23=DSIN(BETA)
      B33=B22
      B32=-B23
      ST(I,J)=A11*B11*S(I,J)+A11*B21*S(I+1,J)+A11*B31*S(I+2,J)+A21*B11*S
     1(I,J+1)+A21*B21*S(I+1,J+1)+A21*B31*S(I+2,J+1)+A31*B11*S(I,J+2)+A31
     2*B21*S(I+1,J+2)+A31*B31*S(I+2,J+2)
      ST(I,J+1)=A12*B11*S(I,J)+A12*B21*S(I+1,J)+A12*B31*S(I+2,J)+A22*B11
     1*S(I,J+1)+A22*B21*S(I+1,J+1)+A22*B31*S(I+2,J+1)+A32*B11*S(I,J+2)+A
     232*B21*S(I+1,J+2)+A32*B31*S(I+2,J+2)
      ST(I,J+2)=A13*B11*S(I,J)+A13*B21*S(I+1,J)+A13*B31*S(I+2,J)+A23*B11
     1*S(I,J+1)+A23*B21*S(I+1,J+1)+A23*B31*S(I+2,J+1)+A33*B11*S(I,J+2)+A
     233*B21*S(I+1,J+2)+A33*B31*S(I+2,J+2)
      ST(I+1,J)=A11*B12*S(I,J)+A11*B22*S(I+1,J)+A11*B32*S(I+2,J)+A21*B12
     1*S(I,J+1)+A21*B22*S(I+1,J+1)+A21*B32*S(I+2,J+1)+A31*B12*S(I,J+2)+A
     231*B22*S(I+1,J+2)+A31*B32*S(I+2,J+2)
      ST(I+1,J+1)=A12*B12*S(I,J)+A12*B22*S(I+1,J)+A12*B32*S(I+2,J)+A22*B
     112*S(I,J+1)+A22*B22*S(I+1,J+1)+A22*B32*S(I+2,J+1)+A32*B12*S(I,J+2)
     2+A32*B22*S(I+1,J+2)+A32*B32*S(I+2,J+2)
      ST(I+1,J+2)=A13*B12*S(I,J)+A13*B22*S(I+1,J)+A13*B32*S(I+2,J)+A23*B
     112*S(I,J+1)+A23*B22*S(I+1,J+1)+A23*B32*S(I+2,J+1)+A33*B12*S(I,J+2)
```

```
     2+A33*B22*S(I+1,J+2)+A33*B32*S(I+2,J+2)
      ST(I+2,J)=A11*B13*S(I,J)+A11*B23*S(I+1,J)+A11*B33*S(I+2,J)+A21*B13
     1*S(I,J+1)+A21*B23*S(I+1,J+1)+A21*B33*S(I+2,J+1)+A31*B13*S(I,J+2)+A
     231*B23*S(I+1,J+2)+A31*B33*S(I+2,J+2)
      ST(I+2,J+1)=A12*B13*S(I,J)+A12*B23*S(I+1,J)+A12*B33*S(I+2,J)+A22*B
     113*S(I,J+1)+A22*B23*S(I+1,J+1)+A22*B33*S(I+2,J+1)+A32*B13*S(I,J+2)
     2+A32*B23*S(I+1,J+2)+A32*B33*S(I+2,J+2)
      ST(I+2,J+2)=A13*B13*S(I,J)+A13*B23*S(I+1,J)+A13*B33*S(I+2,J)+A23*B
     113*S(I,J+1)+A23*B23*S(I+1,J+1)+A23*B33*S(I+2,J+1)+A33*B13*S(I,J+2)
     2+A33*B23*S(I+1,J+2)+A33*B33*S(I+2,J+2)
      ST(J,I)=ST(I,J)
      ST(J+1,I)=ST(I,J+1)
      ST(J+2,I)=ST(I,J+2)
      ST(J,I+1)=ST(I+1,J)
      ST(J+1,I+1)=ST(I+1,J+1)
      ST(J+2,I+1)=ST(I+1,J+2)
      ST(J,I+2)=ST(I+2,J)
      ST(J+1,I+2)=ST(I+2,J+1)
      ST(J+2,I+2)=ST(I+2,J+2)
      GO TO 5
4     ST(I,I)=A11*A11*S(I,I)+2.0D0*A21*A11*S(I+1,I)+2.0D0*A31*A11*S(I+2,
     1I)+A21*A21*S(I+1,I+1)+2.0D0*A31*A21*S(I+2,I+1)+A31*A31*S(I+2,I+2)
      ST(I,I+1)=A12*A11*S(I,I)+A12*A21*S(I+1,I)+A12*A31*S(I+2,I)+A22*A11
     1*S(I,I+1)+A22*A21*S(I+1,I+1)+A22*A31*S(I+2,I+1)+A32*A11*S(I,I+2)+A
     232*A21*S(I+1,I+2)+A32*A31*S(I+2,I+2)
      ST(I,I+2)=A13*A11*S(I,I)+A13*A21*S(I+1,I)+A13*A31*S(I+2,I)+A23*A11
     1*S(I,I+1)+A23*A21*S(I+1,I+1)+A23*A31*S(I+2,I+1)+A33*A11*S(I,I+2)+A
     233*A21*S(I+1,I+2)+A33*A31*S(I+2,I+2)
      ST(I+1,I+1)=A12*A12*S(I,I)+2.0D0*A12*A22*S(I+1,I)+2.0D0*A12*A32*S(
     1I+2,I)+A22*A22*S(I+1,I+1)+2.0D0*A32*A22*S(I+2,I+1)+A32*A32*S(I+2,I
     2+2)
      ST(I+1,I+2)=A13*A12*S(I,I)+A13*A22*S(I+1,I)+A13*A32*S(I+2,I)+A23*A
```

```
     112*S(I,I+1)+A23*A22*S(I+1,J+1)+A23*A32*S(I+2,I+1)+A33*A12*S(I,I+2)
     2+A33*A22*S(I+1,I+2)+A33*A32*S(I+2,I+2)
      ST(I+2,I+2)=A13*A13*S(I,I)+2.0D0*A13*A23*S(I+1,I)+2.0D0*A13*A33*S(
     1I+2,I)+A23*A23*S(I+1,I+1)+2.0D0*A23*A33*S(I+2,I+1)+A33*A33*S(I+2,I
     2+2)
      ST(I+1,I)=ST(I,I+1)
      ST(I+2,I)=ST(I,I+2)
      ST(I+2,I+1)=ST(I+1,I+2)
    5 CONTINUE
    6 CONTINUE
      DO 9 I=1,60,3
      IF (IDOF(I).GT.0) GO TO 9
      DO 8 J=1,3
      JJ=I+J-1
      DO 7 K=1,60
      S(K,JJ)=ST(K,JJ)
      S(JJ,K)=S(K,JJ)
    7 CONTINUE
    8 IDOF(JJ)=-IDOF(JJ)
    9 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE USOL (A,B,MAXB,NEQB,MB,LL,NBLOCK,NSB,NORG,NBKS,NT1,NT2,
     1NRST)
C
C     GENERAL EQUATION SOLVER FOR POSITIVE DEFINITE SYSTEMS WITH
C     PRACTICALLY NO LIMIT ON THE NUMBER OF EQUATIONS, BANDWIDTH
C     OR NUMBER OF SOLUTION VECTORS.
C     SESM PROGRAMS, UNIVERSITY OF CALIFORNIA, BERKELEY
C
      IMPLICIT REAL * 8 (A-G,O-Z)
      DIMENSION A(NSB), B(NSB), MAXB(NEQB)
      CALL TIMEON
      NC=MB+LL
      NBR=(MB-1)/NEQB+1
      INC=NEQB-1
      NMB=NEQB*MB
      N2=NT2
      N1=NT1
      REWIND NORG
      REWIND NBKS
C
C     REDUCE EQUATIONS BLOCK-BY-BLOCK
C
      DO 19 N=1,NBLOCK
      IF (N.GT.1.AND.NBR.EQ.1) GO TO 2
      IF (NBR.EQ.1) GO TO 1
      REWIND N1
      REWIND N2
1     NI=N1
      IF (N.EQ.1) NI=NORG
      READ (NI) A
2     DO 11 I=1,NEQB
      D=A(I)
```

```
      IF (D) 3,11,5
3     M=NEQB*(N-1)+I
      WRITE (6,4) M,D
4     FORMAT (33H0SET OF EQUATIONS MAY BE SINGULAR/26H DIAGONAL TERM OF
     1EQUATION,I8,8H  EQUALS,1PD12.4)
5     II=I
      DO 6 J=2,NC
      II=II+NEQB
6     A(II)=A(II)/D
      DO 7 J=I,NMB,NEQB
      IF (A(J).NE.0.D0) MAXB(I)=J
7     CONTINUE
      JL=I+1
      IF (JL.GT.NEQB) GO TO 11
      II=I
      DO 10 J=JL,NEQB
      II=II+NEQB
      IF (II.GT.NMB) GO TO 10
      C=A(II)
      IF (C.EQ.0.D0) GO TO 10
      C=C*A(I)
      KK=J
      MAX=MAXB(I)
      DO 8 JJ=II,MAX,NEQB
      A(KK)=A(KK)-C*A(JJ)
8     KK=KK+NEQB
      KK=J+NMB
      JJ=I+NMB
      DO 9 L=1,LL
      A(KK)=A(KK)-C*A(JJ)
      KK=KK+NEQB
9     JJ=JJ+NEQB
```

```
10    CONTINUE
11    CONTINUE
C
C     SUBSTITUTE INTO REMAINING EQUATIONS
C
      WRITE (NBKS) A,MAXB
      DO 18 NN=1,NBR
      IF (N+NN.GT.NBLOCK) GO TO 18
      NI=N1
      IF (N.EQ.1) NI=NORG
      IF (NN.EQ.NBR) NI=NORG
      READ (NI) B
      IL=1+NN*NEQB*NEQB
      DO 15 I=1,NEQB
      II=IL
      DO 14 K=1,NEQB
      IF (II.GT.NMB) GO TO 14
      C=A(II)
      IF (C.EQ.0.D0) GO TO 14
      C=C*A(K)
      MAX=MAXB(K)
      KK=I
      DO 12 JJ=II,MAX,NEQB
      B(KK)=B(KK)-C*A(JJ)
12    KK=KK+NEQB
      KK=I+NMB
      JJ=K+NMB
      DO 13 L=1,LL
      B(KK)=B(KK)-C*A(JJ)
      KK=KK+NEQB
13    JJ=JJ+NEQB
14    II=II-INC
```

```
15    IL=IL+NEQB
      IF (NBR.NE.1) GO TO 17
      DO 16 I=1,NSB
16    A(I)=B(I)
      GO TO 18
17    WRITE (N2) B
18    CONTINUE
      M=N1
      N1=N2
19    N2=M
      LS=LL*NEQB
      NEB=NEQB*(NBR+1)
      NUM=NBR*NEQB
      MAX=NEB*LL
      DO 20 I=1,MAX
20    B(I)=0.D0
      REWIND NRST
      DO 27 N=1,NBLOCK
      BACKSPACE NBKS
      READ (NBKS) A,MAXB
      BACKSPACE NBKS
      DO 21 L=1,LL
      K=L*NEB
      DO 21 J=1,NUM
      I=K-NEQB
      B(K)=B(I)
21    K=K-1
      I=NMB
      DO 22 L=1,LL
      K=(L-1)*NEB
      DO 22 J=1,NEQB
      I=I+1
```

```
      K=K+1
22    B(K)=A(I)
      DO 25 I=1,NEQB
      J=NEQB+1-I
      MAX=MAXB(J)
      IF (A(J).EQ.0.D0) GO TO 25
      DO 24 L=1,LL
      KK=J+(L-1)*NEB
      JJ=KK+1
      IL=J+NEQB
      C=B(KK)
      DO 23 II=IL,MAX,NEQB
      C=C-A(II)*B(JJ)
23    JJ=JJ+1
24    B(KK)=C
25    CONTINUE
      I=0
      DO 26 L=1,LL
      K=(L-1)*NEB
      DO 26 J=1,NEQB
      K=K+1
      I=I+1
26    A(I)=B(K)
      WRITE (NRST) (A(I),I=1,LS)
27    CONTINUE
      CALL TIMECK (I)
      IK=I/100
      WRITE (6,28) IK
28    FORMAT (//,3X,33HTIME SPENT IN SUBROUTINE USOL WAS,I5,8H SECONDS)
      RETURN
      END
```

```
      SUBROUTINE ELASTR (INEL,IP,SF)
C
C     CALCULATE STRESSES AT ALL INTEGRATION POINTS OF ELEMENT
C
      IMPLICIT REAL * 8 (A-G,O-Z)
      INTEGER CODE
      COMMON /DAT/ CIP(14,3),X(1000),Y(1000),Z(1000),U(1000),V(1000),W(1
     1000),FE(60),EP(14,6),CODE(1000),IX(200,22),ICI(20,20),NUMNP,NUMEL,
     2NUMMAT
      COMMON /LAS/ E(8,8),D(6,6)
      DIMENSION DNXSI(20), DNETA(20), DNZET(20), BB(6,3), BA(6,60), XX(2
     10), YY(20), ZZ(20), DIS(60), SIG(14,7), DB(60)
      DIMENSION ET(6), DSIG(6)
      DO 1 I=1,20
      II=IX(INEL,I)
      XX(I)=X(II)
      YY(I)=Y(II)
      ZZ(I)=Z(II)
1     CONTINUE
      DO 2 J=1,20
      II=IX(INEL,J)
      DIS(3*J-2)=U(II)
      DIS(3*J-1)=V(II)
2     DIS(3*J)=W(II)
      DO 16 J=1,14
      XSI=CIP(J,1)
      ETA=CIP(J,2)
      ZET=CIP(J,3)
      AJ11=0.0D0
      AJ12=0.0D0
      AJ13=0.0D0
      AJ21=0.0D0
```

```
      AJ22=0.0D0
      AJ23=0.0D0
      AJ31=0.0D0
      AJ32=0.0D0
      AJ33=0.0D0
      DO 3 I=1,20
      DNXSI(I)=ICI(2,I)+ETA*ICI(5,I)+ZET*ICI(7,I)+2.0D0*XSI*ICI(8,I)+ETA
     1*ZET*ICI(11,I)+2.0D0*XSI*ETA*ICI(12,I)+2.0D0*XSI*ZET*ICI(13,I)+ETA
     2*ETA*ICI(14,I)+ZET*ZET*ICI(16,I)+2.0D0*XSI*ETA*ZET*ICI(18,I)+ETA*E
     3TA*ZET*ICI(19,I)+ETA*ZET*ZET*ICI(20,I)
      DNETA(I)=ICI(3,I)+XSI*ICI(5,I)+ZET*ICI(6,I)+2.0D0*ETA*ICI(9,I)+XSI
     1*ZET*ICI(11,I)+XSI*XSI*ICI(12,I)+2.0D0*XSI*ETA*ICI(14,I)+2.0D0*ETA
     2*ZET*ICI(15,I)+ZET*ZET*ICI(17,I)+XSI*XSI*ZET*ICI(18,I)+2.0D0*XSI*E
     3TA*ZET*ICI(19,I)+XSI*ZET*ZET*ICI(20,I)
      DNZET(I)=ICI(4,I)+ETA*ICI(6,I)+XSI*ICI(7,I)+2.0D0*ZET*ICI(10,I)+XS
     1I*ETA*ICI(11,I)+XSI*XSI*ICI(13,I)+ETA*ETA*ICI(15,I)+2.0D0*XSI*ZET*
     2ICI(16,I)+2.0D0*ETA*ZET*ICI(17,I)+XSI*ETA*XSI*ICI(18,I)+XSI*ETA*ET
     3A*ICI(19,I)+2.0D0*XSI*ETA*ZET*ICI(20,I)
    3 CONTINUE
      DO 4 I=1,20
      AJ11=AJ11+DNXSI(I)*XX(I)
      AJ12=AJ12+DNXSI(I)*YY(I)
      AJ13=AJ13+DNXSI(I)*ZZ(I)
      AJ21=AJ21+DNETA(I)*XX(I)
      AJ22=AJ22+DNETA(I)*YY(I)
      AJ23=AJ23+DNETA(I)*ZZ(I)
      AJ31=AJ31+DNZET(I)*XX(I)
      AJ32=AJ32+DNZET(I)*YY(I)
      AJ33=AJ33+DNZET(I)*ZZ(I)
    4 CONTINUE
      DJ=AJ11*AJ22*AJ33+AJ12*AJ23*AJ31+AJ13*AJ21*AJ32-AJ12*AJ21*AJ33-AJ1
     11*AJ23*AJ32-AJ13*AJ22*AJ31
```

```
      IF (DABS(DJ).LT.0.1D-20) GO TO 13
      RJ11=1.0D0/DJ*(AJ22*AJ33-AJ23*AJ32)
      RJ22=1.0D0/DJ*(AJ11*AJ33-AJ13*AJ31)
      RJ33=1.0D0/DJ*(AJ11*AJ22-AJ12*AJ21)
      RJ21=1.0D0/DJ*(-AJ21*AJ33+AJ23*AJ31)
      RJ31=1.0D0/DJ*(AJ21*AJ32-AJ22*AJ31)
      RJ32=1.0D0/DJ*(-AJ11*AJ32+AJ12*AJ31)
      RJ12=1.0D0/DJ*(-AJ12*AJ33+AJ13*AJ32)
      RJ13=1.0D0/DJ*(AJ12*AJ23-AJ13*AJ22)
      RJ23=1.0D0/DJ*(-AJ11*AJ23+AJ13*AJ21)
      DO 5 I=1,6
      DO 5 II=1,3
5     BB(I,II)=0.0D0
      DO 7 I=1,20
      DNX=RJ11*DNXSI(I)+RJ12*DNETA(I)+RJ13*DNZET(I)
      DNY=RJ21*DNXSI(I)+RJ22*DNETA(I)+RJ23*DNZET(I)
      DNZ=RJ31*DNXSI(I)+RJ32*DNETA(I)+RJ33*DNZET(I)
      BB(1,1)=DNX
      BB(2,2)=DNY
      BB(3,3)=DNZ
      BB(4,1)=DNY
      BB(4,2)=DNX
      BB(5,2)=DNZ
      BB(5,3)=DNY
      BB(6,1)=DNZ
      BB(6,3)=DNX
      JJ=3*I-2
      K=3*I-1
      L=3*I
C
C     BA(6,60) REPRESENTS B MATRIX FOR INTEGRATION POINT J
C
```

```
      DO 6 M=1,6
      BA(M,JJ)=BB(M,1)
      BA(M,K)=BB(M,2)
      BA(M,L)=BB(M,3)
6     CONTINUE
7     CONTINUE
      IF (INEL.GT.1) GO TO 8
      MTYPE=IX(INEL,21)
      CALL ELAS (MTYPE)
      MTYPEO=MTYPE
8     MTYPE=IX(INEL,21)
      IF (MTYPE.EQ.MTYPEO) GO TO 9
      CALL ELAS (MTYPE)
      MTYPEO=MTYPE
9     CONTINUE
C
C     MATRIX PRODUCT  D B DIS
C
      DO 12 K=1,6
      SIG(J,K)=0.0D0
      DO 11 L=1,60
      DB(L)=0.0D0
      DO 10 M=1,6
10    DB(L)=DB(L)+D(K,M)*BA(M,L)
      SIG(J,K)=SIG(J,K)+DB(L)*DIS(L)
11    CONTINUE
12    CONTINUE
      GO TO 16
13    DO 14 K=1,7
14    SIG(J,K)=0.0D0
      WRITE (6,15) J,INEL
15    FORMAT (/,3X,38HDET. OF JACOBIAN FOR INTEGRATION POINT,I5,16H
```

```
     1OF  ELEMENT,I5,12H     IS ZERO)
16    CONTINUE
C
C     IF INITIAL STRAINS PRESENT AS FOR TEMPERATURE LOADING, SUBTRACT
C     PRODUCT OF D WITH INITIAL STRAINS FROM STRESSES
C
      IF (IX(INEL,22).LT.2) GO TO 22
      DO 17 I=1,6
      ET(I)=0.0D0
      DSIG(I)=0.0D0
17    CONTINUE
      MTYPE=IX(INEL,21)
      DO 18 I=1,3
18    ET(I)=E(3,MTYPE)*E(4,MTYPE)
      DO 20 I=1,6
      DO 19 J=1,6
      DSIG(I)=DSIG(I)+D(I,J)*ET(J)
19    CONTINUE
20    CONTINUE
      DO 21 I=1,14
      DO 21 J=1,6
      SIG(I,J)=SIG(I,J)-DSIG(J)
21    CONTINUE
22    CONTINUE
C
C     FIND EQUIVALENT STRESS
C
      DO 23 I=1,14
      SIG(I,7)=0.0D0
      A=SIG(I,1)-SIG(I,2)
      Q=SIG(I,2)-SIG(I,3)
      C=SIG(I,3)-SIG(I,1)
```

```
      F=SIG(I,4)
      G=SIG(I,5)
      R=SIG(I,6)
      SIGE=0.5D0*(A*A+Q*Q+C*C)+3.0D0*(F*F+G*G+R*R)
      SIG(I,7)=DSQRT(SIGE)
   23 CONTINUE
C
C     PRINT OUT STRESSES
C
      CALL TITLE
      WRITE (6,24) INEL
   24 FORMAT (/,10X,48HSTRESSES AT ALL INTEGRATION POINTS OF ELEMENT NO,
     1I5,/)
      WRITE (6,25)
   25 FORMAT (/,5X,10HINT. POINT,5X,7HSIG(XX),7X,7HSIG(YY),7X,7HSIG(ZZ),
     17X,7HSIG(XY),7X,7HSIG(YZ),7X,7HSIG(ZX),7X,5HSIGEQ,//)
      DO 26 I=1,14
      WRITE (6,27) I,(SIG(I,J),J=1,7)
   26 CONTINUE
   27 FORMAT (5X,I5,4X,7D14.3)
      CALL SCALER (INEL,IP,SF,SIG)
      RETURN
      END
```

```
      SUBROUTINE SCALER (INEL,IP,SF,SIG)
C
C     THIS SUBROUTINE INDICATES BY WHAT FACTOR THE ORIGINAL
C     LOAD CONDITION HAS TO BE MULTIPLIED IN ORDER TO PRODUCE
C     THE START OF YIELD AT ONE OR MORE INTEGRATION POINTS
C     SIMULTANEOUSLY.
C
      IMPLICIT REAL * 8 (A-G,O-Z)
      INTEGER CODE
      COMMON /DAT/ CIP(14,3),X(1000),Y(1000),Z(1000),U(1000),V(1000),W(1
     1000),FE(60),EP(14,6),CODE(1000),IX(200,22),ICI(20,20),NUMNP,NUMEL,
     2NUMMAT
      COMMON /LAS/ E(8,8),D(6,6)
      DIMENSION SIG(14,7)
      DO 1 I=1,14
      MTYPE=IX(INEL,21)
      YY=E(5,MTYPE)
      SIG(I,7)=YY/SIG(I,7)
1     CONTINUE
      DO 4 I=1,14
      IF (I.EQ.1) GO TO 2
      IF (SIG(I,7).GT.SF) GO TO 3
2     CONTINUE
      SF=SIG(I,7)
      IP=I
3     CONTINUE
4     CONTINUE
      RETURN
      END
```

```
C     PROGRAM 2   JOBSTEP 1
C
C     THE LINEAR SOLUTION IS REPEATED FOR THE SCALED ORIGINAL
C     LOADING.
C
      IMPLICIT REAL * 8 (A-G,O-Z)
      REAL * 4 ES,XX,YY,ZZ,CIPS,RINC
      INTEGER CODE
      COMMON /DAT/ NUMEL,NUMNP,ICI(20,20),CODE(1000),IX(200,22),CIP(14,3
     1),X(1000),Y(1000),Z(1000),U(1000),V(1000),W(1000),AR(3,3)
      COMMON /LAS/ E(8,8),D(6,6)
      COMMON /HAD/ HED(18),IPAGE,LINE
      COMMON /DIS/ XY(1000),YZ(1000),ZX(1000)
      DIMENSION G(3000)
      DIMENSION A1(10000), A2(10000), MAXB(200)
      DIMENSION ES(8,8), XX(1000), YY(1000), ZZ(1000), KODE(1000)
      DIMENSION RINC(10), CIPS(14,3)
      EQUIVALENCE (XX(1),U(1)), (YY(1),V(1)), (ZZ(1),W(1))
      DO 1 I=1,3000
    1 G(I)=0.0D0
      IPAGE=1
      LINE=0
      READ (9) HED
      READ (9) ((ICI(I,J),J=1,20),I=1,20)
      READ (9) ((CIP(I,J),J=1,3),I=1,14)
      READ (9) NUMEL,NUMNP,NUMMAT
      DO 2 I=1,NUMMAT
    2 READ (9) MTYPE,(E(J,MTYPE),J=1,5)
      DO 3 I=1,NUMNP
    3 READ (9) K,CODE(K),X(K),Y(K),Z(K),U(K),V(K),W(K)
      DO 4 I=1,NUMEL
    4 READ (9) M,(IX(M,J),J=1,22)
```

```
      READ (9) NEQB,IBW,NBLOCK
      READ (9) SF,INELO,IPO
      READ (5,5) (RINC(I),I=1,10)
5     FORMAT (8E10.5)
      DO 6 I=1,1000
6     KODE(I)=0
      READ (5,7) NKOD
7     FORMAT (I5)
      IF (NKOD.EQ.0) GO TO 10
      DO 8 I=1,NKOD
      READ (5,9) J,KODE(J)
8     CONTINUE
9     FORMAT (2I5)
10    CONTINUE
      WRITE (6,11) ((ICI(I,J),J=1,20),I=1,20)
11    FORMAT (5X,20I6)
      WRITE (6,12) ((CIP(I,J),J=1,3),I=1,14)
12    FORMAT (10X,3D20.9)
      WRITE (6,13) NUMNP,NUMEL,NUMMAT
13    FORMAT (15X,3I10)
      DO 15 I=1,NUMMAT
      WRITE (6,14) I,(E(J,I),J=1,5)
14    FORMAT (5X,I5,5D15.3)
15    CONTINUE
      DO 17 I=1,NUMNP
      WRITE (6,16) I,CODE(I),X(I),Y(I),Z(I),U(I),V(I),W(I)
16    FORMAT (10X,2I5,6D15.3)
17    CONTINUE
      DO 19 J=1,NUMEL
      WRITE (6,18) J,(IX(J,I),I=1,22)
18    FORMAT (5X,23I5)
19    CONTINUE
```

```
      WRITE (6,20) SF
20    FORMAT (///,10X,22HTHE SCALE FACTOR SF IS,D15.5)
      MBAND=IBW
      LL=1
      NSB=(MBAND+LL)*NEQB
      N1=2
      N5=3
      NTDOF=3*NUMNP
      DO 21 I=1,NUMNP
      J=3*I-2
      G(J)=U(I)*SF
      G(J+1)=V(I)*SF
      G(J+2)=W(I)*SF
21    CONTINUE
      DO 22 I=1,NUMMAT
22    E(4,I)=E(4,I)*SF
      CALL RESOL (A1,A2,G,MAXB,NTDOF,NEQB,MBAND,LL,NBLOCK,NSB,N1,N5)
      REWIND 3
      NNP=NEQB/3
      DO 24 I=1,NBLOCK
      READ (3) ((XY(J),YZ(J),ZX(J)),J=1,NNP)
      IF (I.EQ.NBLOCK) GO TO 24
      NA=(NBLOCK-I)*NNP+1
      DO 23 K=1,NNP
      KK=NA+K-1
      XY(KK)=XY(K)
      YZ(KK)=YZ(K)
23    ZX(KK)=ZX(K)
24    CONTINUE
      CALL TITLE
      WRITE (6,25)
25    FORMAT (1H0,10X,4HNODE,7X,1HX,9X,1HY,9X,1HZ,13X,7HX-DISPL,13X,7HY-
```

```
     1DISPL,13X,7HZ-DISPL)
      DO 29 I=1,NUMNP
      LINE=LINE+1
      WRITE (6,26) I,XY(I),YZ(I),ZX(I)
 26   FORMAT (11X,I3,31X,3D20.8)
      IF (LINE.GE.50) GO TO 27
      GO TO 28
 27   CALL TITLE
      WRITE (6,25)
 28   CONTINUE
 29   CONTINUE
      DO 30 I=1,NUMNP
      IF (CODE(I).LE.7) GO TO 30
      VV=YZ(I)
      WW=ZX(I)
      TANALF=1.7320508D0/2.0D0
      ALFA=DATAN(TANALF)
      AR22=DCOS(ALFA)
      AR33=AR22
      AR23=-DSIN(ALFA)
      AR32=-AR23
      YZ(I)=VV*AR22-WW*AR23
      ZX(I)=-VV*AR32+WW*AR33
 30   CONTINUE
      DO 31 I=1,NUMEL
      INEL=I
      CALL ELASTR (INEL)
 31   CONTINUE
      DO 32 I=1,14
      DO 32 J=1,3
 32   CIPS(I,J)=CIP(I,J)
      DO 33 I=1,NUMMAT
```

```
      DO 33 J=1,5
33    ES(J,I)=E(J,I)
      REWIND 9
      WRITE (9) HED
      WRITE (9) ((ICI(I,J),J=1,20),I=1,20)
      WRITE (9) ((CIPS(I,J),J=1,3),I=1,14)
      WRITE (9) NUMNP,NUMEL,NUMMAT,NEQB,MBAND,NBLOCK,NKOD
      DO 34 I=1,NUMMAT
34    WRITE (9) I,(ES(J,I),J=1,5)
      DO 35 I=1,NUMNP
      XX(I)=X(I)
      YY(I)=Y(I)
      ZZ(I)=Z(I)
35    WRITE (9) I,CODE(I),XX(I),YY(I),ZZ(I)
      DO 36 I=1,NUMEL
36    WRITE (9) I,(IX(I,J),J=1,22)
      WRITE (9) (RINC(I),I=1,10)
      DO 37 I=1,1000
      IF (KODE(I).EQ.0) GO TO 37
      WRITE (9) I,KODE(I)
37    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ELASTR (INEL)
C
C     CALCULATE STRESSES AT ALL INTEGRATION POINTS OF ELEMENT
C
      IMPLICIT REAL * 8 (A-G,O-Z)
      INTEGER CODE
      COMMON /DAT/ NUMEL,NUMNP,ICI(20,20),CODE(1000),IX(200,22),CIP(14,3
     1),X(1000),Y(1000),Z(1000),U(1000),V(1000),W(1000),AR(3,3)
      COMMON /LAS/ E(8,8),D(6,6)
      COMMON /DIS/ XY(1000),YZ(1000),ZX(1000)
      DIMENSION DNXSI(20), DNETA(20), DNZET(20), BB(6,3), BA(6,60), XX(2
     10), YY(20), ZZ(20), DIS(60), SIG(14,7), DB(60)
      DIMENSION ET(6), DSIG(6)
      DO 1 I=1,20
      II=IX(INEL,I)
      XX(I)=X(II)
      YY(I)=Y(II)
      ZZ(I)=Z(II)
    1 CONTINUE
      DO 2 J=1,20
      II=IX(INEL,J)
      DIS(3*J-2)=XY(II)
      DIS(3*J-1)=YZ(II)
    2 DIS(3*J)=ZX(II)
      DO 16 J=1,14
      XSI=CIP(J,1)
      ETA=CIP(J,2)
      ZET=CIP(J,3)
      AJ11=0.0D0
      AJ12=0.0D0
      AJ13=0.0D0
      AJ21=0.0D0
```

```
      AJ22=0.0D0
      AJ23=0.0D0
      AJ31=0.0D0
      AJ32=0.0D0
      AJ33=0.0D0
      DO 3 I=1,20
      DNXSI(I)=ICI(2,I)+ETA*ICI(5,I)+ZET*ICI(7,I)+2.0D0*XSI*ICI(8,I)+ETA
     1*ZET*ICI(11,I)+2.0D0*XSI*ETA*ICI(12,I)+2.0D0*XSI*ZET*ICI(13,I)+ETA
     2*ETA*ICI(14,I)+ZET*ZET*ICI(16,I)+2.0D0*XSI*ETA*ZET*ICI(18,I)+ETA*E
     3TA*ZET*ICI(19,I)+ETA*ZET*ZET*ICI(20,I)
      DNETA(I)=ICI(3,I)+XSI*ICI(5,I)+ZET*ICI(6,I)+2.0D0*ETA*ICI(9,I)+XSI
     1*ZET*ICI(11,I)+XSI*XSI*ICI(12,I)+2.0D0*XSI*ETA*ICI(14,I)+2.0D0*ETA
     2*ZET*ICI(15,I)+ZET*ZET*ICI(17,I)+XSI*XSI*ZET*ICI(18,I)+2.0D0*XSI*E
     3TA*ZET*ICI(19,I)+XSI*ZET*ZET*ICI(20,I)
      DNZET(I)=ICI(4,I)+ETA*ICI(6,I)+XSI*ICI(7,I)+2.0D0*ZET*ICI(10,I)+XS
     1I*ETA*ICI(11,I)+XSI*XSI*ICI(13,I)+ETA*ETA*ICI(15,I)+2.0D0*XSI*ZET*
     2ICI(16,I)+2.0D0*ETA*ZET*ICI(17,I)+XSI*ETA*XSI*ICI(18,I)+XSI*ETA*ET
     3A*ICI(19,I)+2.0D0*XSI*ETA*ZET*ICI(20,I)
3     CONTINUE
      DO 4 I=1,20
      AJ11=AJ11+DNXSI(I)*XX(I)
      AJ12=AJ12+DNXSI(I)*YY(I)
      AJ13=AJ13+DNXSI(I)*ZZ(I)
      AJ21=AJ21+DNETA(I)*XX(I)
      AJ22=AJ22+DNETA(I)*YY(I)
      AJ23=AJ23+DNETA(I)*ZZ(I)
      AJ31=AJ31+DNZET(I)*XX(I)
      AJ32=AJ32+DNZET(I)*YY(I)
      AJ33=AJ33+DNZET(I)*ZZ(I)
4     CONTINUE
      DJ=AJ11*AJ22*AJ33+AJ12*AJ23*AJ31+AJ13*AJ21*AJ32-AJ12*AJ21*AJ33-AJ1
     11*AJ23*AJ32-AJ13*AJ22*AJ31
```

```
      IF (DABS(DJ).LT.0.1D-20) GO TO 13
      RJ11=1.0D0/DJ*(AJ22*AJ33-AJ23*AJ32)
      RJ22=1.0D0/DJ*(AJ11*AJ33-AJ13*AJ31)
      RJ33=1.0D0/DJ*(AJ11*AJ22-AJ12*AJ21)
      RJ21=1.0D0/DJ*(-AJ21*AJ33+AJ23*AJ31)
      RJ31=1.0D0/DJ*(AJ21*AJ32-AJ22*AJ31)
      RJ32=1.0D0/DJ*(-AJ11*AJ32+AJ12*AJ31)
      RJ12=1.0D0/DJ*(-AJ12*AJ33+AJ13*AJ32)
      RJ13=1.0D0/DJ*(AJ12*AJ23-AJ13*AJ22)
      RJ23=1.0D0/DJ*(-AJ11*AJ23+AJ13*AJ21)
      DO 5 I=1,6
      DO 5 II=1,3
    5 BB(I,II)=0.0D0
      DO 7 I=1,20
      DNX=RJ11*DNXSI(I)+RJ12*DNETA(I)+RJ13*DNZET(I)
      DNY=RJ21*DNXSI(I)+RJ22*DNETA(I)+RJ23*DNZET(I)
      DNZ=RJ31*DNXSI(I)+RJ32*DNETA(I)+RJ33*DNZET(I)
      BB(1,1)=DNX
      BB(2,2)=DNY
      BB(3,3)=DNZ
      BB(4,1)=DNY
      BB(4,2)=DNX
      BB(5,2)=DNZ
      BB(5,3)=DNY
      BB(6,1)=DNZ
      BB(6,3)=DNX
      JJ=3*I-2
      K=3*I-1
      L=3*I
      DO 6 M=1,6
      BA(M,JJ)=BB(M,1)
      BA(M,K)=BB(M,2)
```

```
      BA(M,L)=BB(M,3)
6     CONTINUE
7     CONTINUE
      IF (INEL.GT.1) GO TO 8
      MTYPE=IX(INEL,21)
      CALL ELAS (MTYPE)
      MTYPEO=MTYPE
8     MTYPE=IX(INEL,21)
      IF (MTYPE.EQ.MTYPEO) GO TO 9
      CALL ELAS (MTYPE)
      MTYPEO=MTYPE
9     CONTINUE
      DO 12 K=1,6
      SIG(J,K)=0.0D0
      DO 11 L=1,60
      DB(L)=0.0D0
      DO 10 M=1,6
10    DB(L)=DB(L)+D(K,M)*BA(M,L)
      SIG(J,K)=SIG(J,K)+DB(L)*DIS(L)
11    CONTINUE
12    CONTINUE
      GO TO 16
13    DO 14 K=1,7
14    SIG(J,K)=0.0D0
      WRITE (6,15) J,INEL
15    FORMAT (/,3X,38HDET. OF JACOBIAN FOR INTEGRATION POINT,I5,16H
     1OF  ELEMENT,I5,12H      IS ZERO)
16    CONTINUE
      IF (IX(INEL,22).LT.2) GO TO 22
      DO 17 I=1,6
      ET(I)=0.0D0
      DSIG(I)=0.0D0
```

```
17     CONTINUE
       MTYPE=IX(INEL,21)
       DO 18 I=1,3
18     ET(I)=E(3,MTYPE)*E(4,MTYPE)
       DO 20 I=1,6
       DO 19 J=1,6
       DSIG(I)=DSIG(I)+D(I,J)*ET(J)
19     CONTINUE
20     CONTINUE
       DO 21 I=1,14
       DO 21 J=1,6
       SIG(I,J)=SIG(I,J)-DSIG(J)
21     CONTINUE
22     CONTINUE
       DO 23 I=1,14
       SIG(I,7)=0.0D0
       A=SIG(I,1)-SIG(I,2)
       Q=SIG(I,2)-SIG(I,3)
       C=SIG(I,3)-SIG(I,1)
       F=SIG(I,4)
       G=SIG(I,5)
       R=SIG(I,6)
       SIGE=0.5D0*(A*A+Q*Q+C*C)+3.0D0*(F*F+G*G+R*R)
       SIG(I,7)=DSQRT(SIGE)
23     CONTINUE
C
C      PRINT OUT STRESSES
C
       CALL TITLE
       WRITE (6,24) INEL
24     FORMAT (/,10X,48HSTRESSES AT ALL INTEGRATION POINTS OF ELEMENT NO,
      1I5,/)
```

```
      WRITE (6,25)
25    FORMAT (/,5X,10HINT. POINT,5X,7HSIG(XX),7X,7HSIG(YY),7X,7HSIG(ZZ),
     17X,7HSIG(XY),7X,7HSIG(YZ),7X,7HSIG(ZX),7X,5HSIGEQ,//)
      DO 26 I=1,14
      WRITE (6,27) I,(SIG(I,J),J=1,7)
26    CONTINUE
27    FORMAT (5X,I5,4X,7D14.3)
      CALL RITOFF (INEL,SIG)
      RETURN
      END
```

```
      SUBROUTINE TITLE
      COMMON /HAD/ HED(18),IPAGE,LINE
      WRITE (6,1) IPAGE
1     FORMAT (1H1,3X,39H60-DCF GENERAL HEXAHEDRON ELEMENT, HATT,50X,4HPA
     1GE,I3)
      WRITE (6,2) HED
2     FORMAT (1H0,10X,18A4)
      IPAGE=IPAGE+1
      LINE=0
      RETURN
      END
```

```
      SUBROUTINE RESOL (A,B,G,MAXB,NTDOF,NEQB,MB,LL,NBLOCK,NSB,NORG,NRST
     1)
C
C     MODIFIED SOLUTION ROUTINE FROM USOL. THE TRIANGULIZED
C     EQUATIONS RESIDE ON UNIT 2. THE LOAD VECTOR G HAS TO BE
C     MODIFIED FIRST. THEN THE BACKSUBSTITUTION CAN BE PERFORMED.
C
      IMPLICIT REAL * 8 (A-G,O-Z)
      DIMENSION A(NSB), B(NSB), G(NTDOF), MAXB(NEQB)
      CALL TIMEON
      NBR=(MB-1)/NEQB+1
      NMB=NEQB*MB
      DO 5 N=1,NBLOCK
      READ (NORG) A,MAXB
      DO 2 I=1,NEQB
      JJ=NEQB+I
      JK=MAXB(I)
      C=A(I)
      DO 1 J=JJ,JK,NEQB
      A(J)=A(J)*C
    1 CONTINUE
    2 CONTINUE
      DO 4 I=1,NEQB
      NN=(N-1)*NEQB+I
      C=A(I)
      G(NN)=G(NN)/C
      IF (NN.EQ.NTDOF) GO TO 5
      NK=NN+MB
      JJ=0
      NS=NN+1
      DO 3 J=NS,NK
      KK=I+NEQB+JJ*NEQB
```

```
      IF (KK.GT.MAXB(I)) GO TO 4
      G(J)=G(J)-G(NN)*A(KK)
      JJ=JJ+1
    3 CONTINUE
    4 CONTINUE
    5 CONTINUE
      LS=LL*NEQB
      NEB=NEQB*(NBR+1)
      NUM=NBR*NEQB
      MAX=NEB*LL
      DO 6 I=1,MAX
    6 B(I)=0.D0
      REWIND NRST
      DO 14 N=1,NBLOCK
      MI=NORG
      BACKSPACE MI
      READ (MI) A,MAXB
      DO 7 II=1,NEQB
      IL=NSB-NEQB+II
      IM=(NBLOCK-N)*NEQB+II
      IF (IM.GT.NTDOF) GO TO 7
      A(IL)=G(IM)
    7 CONTINUE
      BACKSPACE MI
      DO 8 L=1,LL
      K=L*NEB
      DO 8 J=1,NUM
      I=K-NEQB
      B(K)=B(I)
    8 K=K-1
      I=NMB
      DO 9 L=1,LL
```

```
      K=(L-1)*NEB
      DO 9 J=1,NEQB
      I=I+1
      K=K+1
      B(K)=A(I)
9     CONTINUE
      DO 12 I=1,NEQB
      J=NEQB+1-I
      MAX=MAXB(J)
      IF (A(J).EQ.0.D0) GO TO 12
      DO 11 L=1,LL
      KK=J+(L-1)*NEB
      JJ=KK+1
      IL=J+NEQB
      C=B(KK)
      DO 10 II=IL,MAX,NEQB
      C=C-A(II)*B(JJ)
10    JJ=JJ+1
11    B(KK)=C
12    CONTINUE
      I=0
      DO 13 L=1,LL
      K=(L-1)*NEB
      DO 13 J=1,NEQB
      K=K+1
      I=I+1
13    A(I)=B(K)
      WRITE (NRST) (A(I),I=1,LS)
14    CONTINUE
      CALL TIMECK (I)
      IK=I/100
      WRITE (6,15) IK
```

```
15     FORMAT (//,3X,34HTIME SPENT IN SUBROUTINE RESOL WAS,I5,8H SECONDS)    D  97
       RETURN                                                                D  98
       END                                                                   D  99
```

```
      SUBROUTINE ELAS (MTYPE)
C
C     CALCULATION OF ELASTICITY MATRIX (D)
C
      IMPLICIT REAL * 8 (A-G,O-Z)
      COMMON /LAS/ E(8,8),D(6,6)
      EE=E(1,MTYPE)
      XNU=E(2,MTYPE)
      DO 1 I=1,6
      DO 1 J=1,6
1     D(I,J)=0.0D0
      CONST=EE*(1.0D0-XNU)/((1.0D0+XNU)*(1.0D0-2.0D0*XNU))
      D(1,1)=CONST
      D(1,2)=CONST*XNU/(1.0D0-XNU)
      D(1,3)=D(1,2)
      D(2,2)=CONST
      D(2,3)=D(1,2)
      D(3,3)=CONST
      D(4,4)=CONST*(1.0D0-2.0D0*XNU)/(2.0D0*(1.0D0-XNU))
      D(5,5)=D(4,4)
      D(6,6)=D(4,4)
      D(2,1)=D(1,2)
      D(3,1)=D(1,3)
      D(3,2)=D(2,3)
      RETURN
      END
```

```
      SUBROUTINE RITOFF (INEL,SIG)
C
C     LINEAR DISPLACEMENTS, LINEAR STRESSES AT ALL INTEGRATION POINTS,
C     ARE STORED ELEMENT-BY-ELEMENT ON UNIT 1 .
C
      IMPLICIT REAL * 8 (A-G,O-Z)
      REAL * 4 DISS,SIGS
      INTEGER CODE
      COMMON /DAT/ NUMEL,NUMNP,ICI(20,20),CODE(1000),IX(200,22),CIP(14,3
     1),X(1000),Y(1000),Z(1000),U(1000),V(1000),W(1000),AR(3,3)
      COMMON /DIS/ XY(1000),YZ(1000),ZX(1000)
      DIMENSION SIG(14,7)
      DIMENSION DISS(60), SIGS(14,6)
      DO 1 I=1,14
      DO 1 J=1,6
1     SIGS(I,J)=SIG(I,J)
      DO 2 I=1,20
      M=IX(INEL,I)
      II=3*I-2
      IJ=3*I-1
      IK=3*I
      DISS(II)=XY(M)
      DISS(IJ)=YZ(M)
2     DISS(IK)=ZX(M)
      WRITE (1) (DISS(I),I=1,60),((SIGS(I,J),J=1,6),I=1,14)
      RETURN
      END
```

```
C      PROGRAM 2   JOBSTEP 2.
C
C      ITERATIVE SOLUTION FOR NONLINEAR RANGE OF PROBLEM.
C      CONSTANT STIFFNESS, INITIAL STRESS
C
       INTEGER CODE
       REAL * 8 GD,XYZ,A1,A2
       COMMON /DAT/ X(1000),Y(1000),Z(1000),CIP(14,3),ICI(20,20),CODE(100
      10),IX(200,22),KODE(1000),NUMNP,NUMEL
       COMMON /HAD/ HED(18),IPAGE,LINE
       COMMON /GEE/ G(3000)
       COMMON /DEL/ XY(1000),YZ(1000),ZX(1000)
       COMMON /LAS/ E(8,8),D(6,6)
       COMMON /STR/ SIG(350,6),EUP(350)
       COMMON /GEO/ GE(3000)
       DIMENSION A1(10000), A2(10000), GD(3000), MAXB(200)
       DIMENSION SIGIN(14,6), DSIG(14,6), DIS(60), ELSTR(14,6), SIGM(6),
      1ODIS(60), OSTR(14,6), RINC(15)
       DIMENSION XYZ(3,50)
       DIMENSION DIST(3000)
       READ (5,1) NJOB
1      FORMAT (I5)
       DO 2 I=1,1000
2      KODE(I)=0
       ELTIME=0.0
C
C      READ PROBLEM DATA
C
       READ (9) HED
       READ (9) ((ICI(I,J),J=1,20),I=1,20)
       READ (9) ((CIP(I,J),J=1,3),I=1,14)
       READ (9) NUMNP,NUMEL,NUMMAT,NEQB,MBAND,NBLOCK,NKOD
```

```
      DO 3 I=1,NUMMAT
3     READ (9) M,(E(J,M),J=1,5)
      DO 4 I=1,NUMNP
4     READ (9) M,CODE(M),X(M),Y(M),Z(M)
      DO 5 I=1,NUMEL
5     READ (9) M,(IX(M,J),J=1,22)
      READ (9) (RINC(I),I=1,10)
      IF (NKOD.EQ.0) GO TO 7
      DO 6 I=1,NKOD
6     READ (9) J,KODE(J)
7     CONTINUE
C
C     CONSTANT PARAMETERS OF PROBLEM
C
      LL=1
      NSB=(MBAND+LL)*NEQB
      N1=2
      N5=3
      NTDOF=3*NUMNP
      ICOLL=0
      IPAGE=1
      LINE=0
      MAXIT=25
      NPINT=14*NUMEL
      IF (NJOB.EQ.1) GO TO 9
      DO 8 I=1,NUMEL
8     READ (1) (ODIS(K),K=1,60),((OSTR(K,L),L=1,6),K=1,14)
      READ (9) NSTART
      READ (9) ((SIG(I,J),J=1,6),I=1,NPINT),(EUP(I),I=1,NPINT)
      GO TO 11
9     CONTINUE
      NSTART=1
```

```
      DO 10 I=1,NPINT
10    EUP(I)=0.0
11    CONTINUE
C
C     DO - LOOP 200. THE NUMBER OF THE LOAD INCREMENT REPRESENTS THE
C     PARAMETER I.
C
      DO 50 I=NSTART,15
      CALL TIMEON
      NINC=I-NSTART
      IF (NINC.EQ.0) GO TO 12
      ELTIME=ELTIME+DTIME
      INCR=I
      IF (ELTIME.GT.600.) GO TO 51
12    CONTINUE
      REWIND 1
      DO 13 II=1,3000
13    G(II)=0.0
      DO 24 J=1,NUMEL
C
C     READ FROM UNIT 1. LINEAR DISPLACEMENT / LINEAR STRESS
C     AND FIND LINEAR DISPLACEMENT INCREMENT / LINEAR STRESS
C     INCREMENT ACCORDING TO LOAD INCREMENT.
C
      READ (1) (ODIS(K),K=1,60),((OSTR(K,L),L=1,6),K=1,14)
      DO 14 K=1,20
      DIS(3*K-2)=RINC(I)*ODIS(3*K-2)
      DIS(3*K-1)=RINC(I)*ODIS(3*K-1)
      DIS(3*K)=RINC(I)*ODIS(3*K)
      N=IX(J,K)
      DIST(3*N-2)=DIS(3*K-2)
      DIST(3*N-1)=DIS(3*K-1)
```

```
14    DIST(3*N)=DIS(3*K)
      DO 15 K=1,14
      DO 15 L=1,6
15    ELSTR(K,L)=RINC(I)*OSTR(K,L)
      IF (I.GT.1) GO TO 17
C
C     FOR FIRST LOAD INCREMENT, THE UPDATED TOTAL STRESS IS SET
C     EQUAL TO LINEAR STRESS FROM JOBSTEP 1.
C
      DO 16 K=1,14
      DO 16 L=1,6
      KK=(J-1)*14+K
16    SIG(KK,L)=OSTR(K,L)
17    CONTINUE
      DO 19 K=1,14
      DO 18 L=1,6
      KK=(J-1)*14+K
      SIGM(L)=SIG(KK,L)+ELSTR(K,L)
18    CONTINUE
C
C     CHECK YIELD CRITERION FOR SIG + EL. INCR. OF SIG.
C
      CALL SIGEQU (SIGM,SIGEQ)
      EPSUP=EUP(KK)
      INEL=J
      CALL STRSTR (INEL,EPSUP,YS,A,ICOLL)
      F1=SIGEQ-YS
      IF (F1.GT..1E-10) GO TO 21
19    CONTINUE
C
C     IF NONE OF THE INTEGRATION POINTS IS YIELDING, UPDATE
C     TOTAL SIG.
```

```
C
      DO 20 K=1,14
      DO 20 L=1,6
      KK=(J-1)*14+K
      SIG(KK,L)=SIG(KK,L)+ELSTR(K,L)
20    CONTINUE
      GO TO 23
21    CONTINUE
C
C     LINEAR STRESS INCREMENT TRANSFERRED TO SR PLASTR.
C
      DO 22 K=1,14
      DO 22 L=1,6
      DSIG(K,L)=ELSTR(K,L)
22    CONTINUE
      CALL PLASTR (INEL,DSIG,SIGIN,ICOLL)
C
C     FIND RESIDUAL FORCES FOM INITIAL STRESSES.
C
      CALL EQUFOR (INEL,SIGIN,NTDOF)
23    CONTINUE
24    CONTINUE
C
C     TRANSFORM EQUIVALENT FORCE VECTOR G TO DOUBLE PRECISION GD.
C     SET GE EQUAL TO G.
C
      DO 25 K=1,NTDOF
      GD(K)=G(K)
25    GE(K)=G(K)
C
C     FIND GMAX FROM RESIDUAL FORCE VECTOR G(I).
C
```

```
      GMAX=ABS(G(1))
      IND=1
      DO 27 J=2,NTDOF
      ABSG=ABS(G(J))
      IF (ABSG.GT.GMAX) GO TO 26
      GO TO 27
26    GMAX=ABSG
      IND=J
27    CONTINUE
      WRITE (6,28) GMAX,G(IND)
28    FORMAT (//,10X,7HGMAX = ,E15.4,11H  G(IND) = ,E15.4)
C
C     DO - LOOP 205. THE NUMBER OF THE CURRENT ITERATION REPRESENTS
C     THE PARAMETER J.
C
      DO 49 J=1,MAXIT
      IF (J.EQ.1) GO TO 29
C
C     TEST FOR MAX NUMBER OF ITERATIONS.
C
      IF (J.EQ.25) GO TO 43
C
C     TEST FOR CONVERGENCE BY COMPARING G(IND) TO GMAX.
C
      GCMAX=0.05*GMAX
      IF (GIND.LT.GCMAX) GO TO 45
29    CONTINUE
      DO 30 JJ=1,3000
30    G(JJ)=0.0
      NTDOF=3*NUMNP
      LL=1
      NSB=(MBAND+LL)*NEQB
```

```
      N1=2
      N5=3
C
C     SOLVE FOR DISPLACEMENTS WITH EQUIVALENT FORCES AS LOAD VECTOR.
C
      CALL RESOL (A1,A2,GD,MAXB,NTDOF,NEQB,MBAND,LL,NBLOCK,NSB,N1,N5)
      REWIND 3
C
C     RETRIEVE DISPLACEMENTS FROM UNIT 3. XYZ IN DOUBLE PRECISION.
C     XY,YZ,ZX IN SINGLE PRECISION.
C
      NNP=NEQB/3
      DO 32 K=1,NBLOCK
      READ (3) ((XYZ(L,M),L=1,3),M=1,NNP)
      NA=(NBLOCK-K)*NNP+1
      DO 31 L=1,NNP
      LL=NA+L-1
      XY(LL)=XYZ(1,L)
      YZ(LL)=XYZ(2,L)
31    ZX(LL)=XYZ(3,L)
32    CONTINUE
      NUMIT=J
C
C     TRANSFER DISPLACEMENTS TO ACCELERATION ROUTINE.
C
      CALL ACCEL (DIST,XY,YZ,ZX,AVG,NUMIT)
C
C     RETURN DISPLACEMENTS TO GLOBAL SYSTEM.
C
      DO 33 JI=1,NUMNP
      IF (CODE(JI).LE.7) GO TO 33
      VV=YZ(JI)
```

```
      WW=ZX(JI)
      TANALF=1.732051/2.0
      ALFA=ATAN(TANALF)
      AR22=COS(ALFA)
      AR33=AR22
      AR23=-SIN(ALFA)
      AR32=-AR23
      YZ(JI)=VV*AR22-WW*AR23
      ZX(JI)=-VV*AR32+WW*AR33
   33 CONTINUE
      REWIND 1
      DO 41 K=1,NUMEL
      INEL=K
C
C     ASSIGN DISPLACEMENTS TO EVERY ELEMENT.
C
      DO 34 L=1,20
      LL=IX(K,L)
      DIS(3*L-2)=XY(LL)
      DIS(3*L-1)=YZ(LL)
      DIS(3*L)=ZX(LL)
   34 CONTINUE
C
C     FIND ELASTIC STRESS INCREMENTS CORRESPONDING TO THE DISPLACEMENT
C     INCREMENT FROM SR RESOL.
C
      CALL RELSTR (INEL,DIS,ELSTR)
      DO 36 L=1,14
      DO 35 M=1,6
      KK=(K-1)*14+L
      SIGM(M)=SIG(KK,M)+ELSTR(L,M)
   35 CONTINUE
```

```
C
C     CHECK YIELD CRITERION FOR SIG+EL. INCR. OF SIG.
C
      CALL SIGEQU (SIGM,SIGEQ)
      EPSUP=EUP(KK)
      CALL STRSTR (INEL,EPSUP,YS,A,ICOLL)
      F1=SIGEQ-YS
      IF (F1.GT..1E-10) GO TO 38
36    CONTINUE
C
C     IF NONE OF THE INTEGRATION POINTS IS YIELDING, UPDATE TOTAL SIG.
C
      DO 37 L=1,14
      DO 37 M=1,6
      KK=(K-1)*14+L
      SIG(KK,M)=SIG(KK,M)+ELSTR(L,M)
37    CONTINUE
      GO TO 40
38    CONTINUE
C
C     LINEAR STRESS INCREMENT TRANSFERRED TO SR PLASTR.
C
      DO 39 L=1,14
      DO 39 M=1,6
      DSIG(L,M)=ELSTR(L,M)
39    CONTINUE
      CALL PLASTR (INEL,DSIG,SIGIN,ICOLL)
C
C     FIND RESIDUAL FORCES FROM INITIAL STRESSES.
C
      CALL EQUFOR (INEL,SIGIN,NTDOF)
      NTDOF=3*NUMNP
```

```
40    CONTINUE
41    CONTINUE
C
C     CALL LOAD ROUTINE TO COMPENSATE THE RESIDUAL LOAD VECTOR TO
C     THE UNIFORM ACCELERATION.
C
      CALL LOAD (GD,AVG,NTDOF)
      GINN=GD(IND)
      GIND=ABS(GINN)
      WRITE (6,42) GIND,J,I,GD(IND)
42    FORMAT (//,10X,9HG(IND) = ,E15.4,17H  ITERATION NR = ,I5,14H  LOAD
     1 INCR = ,I5,12H  GD(IND) = ,D15.4)
      GO TO 49
43    CONTINUE
      CALL TITLE
      WRITE (6,44) I
44    FORMAT (//,5X,45HCONVERGENCE NOT ACHIEVED WITHIN 25 ITERATIONS,//,
     15X,20HLOAD INCREMENT NO = ,I5)
      CALL PRINTS (NUMEL)
      GO TO 51
45    CONTINUE
      CALL TITLE
      WRITE (6,46) J,I
46    FORMAT (//,5X,43HCONVERGENCE ACHIEVED DURING ITERATION NR = ,I5,20
     1HLOAD INCREMENT NR = ,I5)
      CALL PRINTS (NUMEL)
      CALL TIMECK (III)
      DTIME=III/100.
      WRITE (6,47) I,DTIME
47    FORMAT (//,5X,33HELAPSED TIME DURING LOADSTEP NR =,I5,5H  WAS,F15.
     12,9H  SECONDS)
      IF (ICOLL.EQ.0) GO TO 50
```

```
      WRITE (6,48)
48    FORMAT (//,5X,26HCOLLAPSE STRAIN IS REACHED,//)
      CALL PRINTS (NUMEL)
      GO TO 51
49    CONTINUE
50    CONTINUE
51    CONTINUE
C
C     WRITE PROBLEM DATA, UPDATED STRESSES, AND UNIAXIAL PLASTIC
C     STRAINS ON UNIT 9 FOR REFERENCE IN THE NEXT JOB.
C
      REWIND 9
      WRITE (9) HED
      WRITE (9) ((ICI(I,J),J=1,20),I=1,20)
      WRITE (9) ((CIP(I,J),J=1,3),I=1,14)
      WRITE (9) NUMNP,NUMEL,NUMMAT,NEQB,MBAND,NBLOCK,NKCO
      DO 52 I=1,NUMMAT
52    WRITE (9) I,(E(J,I),J=1,5)
      DO 53 I=1,NUMNP
53    WRITE (9) I,CODE(I),X(I),Y(I),Z(I)
      DO 54 I=1,NUMEL
54    WRITE (9) I,(IX(I,J),J=1,22)
      WRITE (9) (RINC(I),I=1,10)
      DO 55 I=1,NUMNP
      IF (KODE(I).EQ.0) GO TO 55
      WRITE (9) I,KODE(I)
55    CONTINUE
      WRITE (9) INCR
      WRITE (9) ((SIG(I,J),J=1,6),I=1,NPINT),(EUP(I),I=1,NPINT)
      RETURN
      END
```

```
      SUBROUTINE RELSTR (INEL,DIS,ELSTR)
C
C     FROM GIVEN DISPLACEMENTS, ELASTIC STRESSES ARE FOUND.
C
      INTEGER CODE
      COMMON /DAT/ X(1000),Y(1000),Z(1000),CIP(14,3),ICI(20,20),CODE(100
     10),IX(200,22),KODE(1000),NUMNP,NUMEL
      COMMON /LAS/ E(8,8),D(6,6)
      DIMENSION DIS(60), ELSTR(14,6), DB(60), DNXSI(20), DNETA(20), DNZE
     1T(20), BB(6,3), BA(6,60), XX(20), YY(20), ZZ(20)
      DO 1 I=1,20
      II=IX(INEL,I)
      XX(I)=X(II)
      YY(I)=Y(II)
      ZZ(I)=Z(II)
1     CONTINUE
      DO 16 J=1,14
      XSI=CIP(J,1)
      ETA=CIP(J,2)
      ZET=CIP(J,3)
      AJ11=0.0
      AJ12=0.0
      AJ13=0.0
      AJ21=0.0
      AJ22=0.0
      AJ23=0.0
      AJ31=0.0
      AJ32=0.0
      AJ33=0.0
      DO 2 I=1,20
      DNXSI(I)=ICI(2,I)+ETA*ICI(5,I)+ZET*ICI(7,I)+2.0*XSI*ICI(8,I)+ETA*Z
     1ET*ICI(11,I)+2.0*XSI*ETA*ICI(12,I)+2.0*XSI*ZET*ICI(13,I)+ETA*ETA*I
```

```
     2CI(14,I)+ZET*ZET*ICI(16,I)+2.0*XSI*ETA*ZET*ICI(18,I)+ETA*ETA*ZET*I
     3CI(19,I)+ETA*ZET*ZET*ICI(20,I)
      DNETA(I)=ICI(3,I)+XSI*ICI(5,I)+ZET*ICI(6,I)+2.0*ETA*ICI(9,I)+XSI*Z
     1ET*ICI(11,I)+XSI*XSI*ICI(12,I)+2.0*XSI*ETA*ICI(14,I)+2.0*ETA*ZET*I
     2CI(15,I)+ZET*ZET*ICI(17,I)+XSI*XSI*ZET*ICI(18,I)+2.0*XSI*ETA*ZET*I
     3CI(19,I)+XSI*ZET*ZET*ICI(20,I)
      DNZET(I)=ICI(4,I)+ETA*ICI(6,I)+XSI*ICI(7,I)+2.0*ZET*ICI(10,I)+XSI*
     1ETA*ICI(11,I)+XSI*XSI*ICI(13,I)+ETA*ETA*ICI(15,I)+2.0*XSI*ZET*ICI(
     216,I)+2.0*ETA*ZET*ICI(17,I)+XSI*ETA*XSI*ICI(18,I)+XSI*ETA*ETA*ICI(
     319,I)+2.0*XSI*ETA*ZET*ICI(20,I)
2     CONTINUE
      DO 3 I=1,20
      AJ11=AJ11+DNXSI(I)*XX(I)
      AJ12=AJ12+DNXSI(I)*YY(I)
      AJ13=AJ13+DNXSI(I)*ZZ(I)
      AJ21=AJ21+DNETA(I)*XX(I)
      AJ22=AJ22+DNETA(I)*YY(I)
      AJ23=AJ23+DNETA(I)*ZZ(I)
      AJ31=AJ31+DNZET(I)*XX(I)
      AJ32=AJ32+DNZET(I)*YY(I)
      AJ33=AJ33+DNZET(I)*ZZ(I)
3     CONTINUE
      DJ=AJ11*AJ22*AJ33+AJ12*AJ23*AJ31+AJ13*AJ21*AJ32-AJ12*AJ21*AJ33-AJ1
     11*AJ23*AJ32-AJ13*AJ22*AJ31
      IF (ABS(DJ).LT.0.1E-16) GO TO 7
      RJ11=1.0/DJ*(AJ22*AJ33-AJ23*AJ32)
      RJ22=1.0/DJ*(AJ11*AJ33-AJ13*AJ31)
      RJ33=1.0/DJ*(AJ11*AJ22-AJ12*AJ21)
      RJ21=1.0/DJ*(-AJ21*AJ33+AJ23*AJ31)
      RJ31=1.0/DJ*(AJ21*AJ32-AJ22*AJ31)
      RJ32=1.0/DJ*(-AJ11*AJ32+AJ12*AJ31)
      RJ12=1.0/DJ*(-AJ12*AJ33+AJ13*AJ32)
```

```
      RJ13=1.0/DJ*(AJ12*AJ23-AJ13*AJ22)
      RJ23=1.0/DJ*(-AJ11*AJ23+AJ13*AJ21)
      DO 4 I=1,6
      DO 4 II=1,3
    4 BB(I,II)=0.0
      DO 6 I=1,20
      DNX=RJ11*DNXSI(I)+RJ12*DNETA(I)+RJ13*DNZET(I)
      DNY=RJ21*DNXSI(I)+RJ22*DNETA(I)+RJ23*DNZET(I)
      DNZ=RJ31*DNXSI(I)+RJ32*DNETA(I)+RJ33*DNZET(I)
      BB(1,1)=DNX
      BB(2,2)=DNY
      BB(3,3)=DNZ
      BB(4,1)=DNY
      BB(4,2)=DNX
      BB(5,2)=DNZ
      BB(5,3)=DNY
      BB(6,1)=DNZ
      BB(6,3)=DNX
      JJ=3*I-2
      K=3*I-1
      L=3*I
      DO 5 M=1,6
      BA(M,JJ)=BB(M,1)
      BA(M,K)=BB(M,2)
      BA(M,L)=BB(M,3)
    5 CONTINUE
    6 CONTINUE
      GO TO 10
    7 WRITE (6,8) J,INEL
    8 FORMAT (//,5X,51HDETERMINANT OF JACOBIAN FOR INTEGRATION POINT NR
     1= ,I5,14H  ELEMENT NR =,I5,9H  IS ZERO)
      DO 9 K=1,6
```

```
      KK=J
9     ELSTR(KK,K)=0.0
      GO TO 16
10    CONTINUE
      IF (INEL.GT.1) GO TO 11
      MTYPE=IX(INEL,21)
      CALL ELAS (MTYPE)
      MTYPEO=MTYPE
11    MTYPE=IX(INEL,21)
      IF (MTYPE.EQ.MTYPEO) GO TO 12
      CALL ELAS (MTYPE)
      MTYPEO=MTYPE
12    CONTINUE
      DO 15 K=1,6
      ELSTR(J,K)=0.0
      DO 14 L=1,60
      DB(L)=0.0
      DO 13 M=1,6
13    DB(L)=DB(L)+D(K,M)*BA(M,L)
      ELSTR(J,K)=ELSTR(J,K)+DB(L)*DIS(L)
14    CONTINUE
15    CONTINUE
16    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EQUFOR (INEL,SIGIN,NTDOF)
C
C     FROM THE INITIAL STRESSES THE EQUIVALENT NODAL FORCES ARE COMPUTED
C     THE CORRESPONDING TOTAL LOAD VECTOR IS UPDATED ELEMENT BY ELEMENT.
C
      INTEGER CCD,CODE
      COMMON /DAT/ X(1000),Y(1000),Z(1000),CIP(14,3),ICI(20,20),CODE(100
     10),IX(200,22),KCDE(1000),NUMNP,NUMEL
      COMMON /GEE/ G(3000)
      DIMENSION DNXSI(20), DNETA(20), DNZET(20), BB(6,3), BA(6,60), DETJ
     1(14), XX(20), YY(20), ZZ(20), SIGIN(14,6), FE(60)
      BC=0.886426593
      CC=0.335180055
      DO 1 I=1,60
1     FE(I)=0.0
      DO 2 I=1,20
      II=IX(INEL,I)
      XX(I)=X(II)
      YY(I)=Y(II)
      ZZ(I)=Z(II)
2     CONTINUE
      DO 13 J=1,14
      XSI=CIP(J,1)
      ETA=CIP(J,2)
      ZET=CIP(J,3)
      AJ11=0.0
      AJ12=0.0
      AJ13=0.0
      AJ21=0.0
      AJ22=0.0
      AJ23=0.0
      AJ31=0.0
```

```
      AJ32=0.0
      AJ33=0.0
      DO 3 I=1,20
      DNXSI(I)=ICI(2,I)+ETA*ICI(5,I)+ZET*ICI(7,I)+2.0*XSI*ICI(8,I)+ETA*Z
     1ET*ICI(11,I)+2.0*XSI*ETA*ICI(12,I)+2.0*XSI*ZET*ICI(13,I)+ETA*ETA*I
     2CI(14,I)+ZET*ZET*ICI(16,I)+2.0*XSI*ETA*ZET*ICI(18,I)+ETA*ETA*ZET*I
     3CI(19,I)+ETA*ZET*ZET*ICI(20,I)
      DNETA(I)=ICI(3,I)+XSI*ICI(5,I)+ZET*ICI(6,I)+2.0*ETA*ICI(9,I)+XSI*Z
     1ET*ICI(11,I)+XSI*XSI*ICI(12,I)+2.0*XSI*ETA*ICI(14,I)+2.0*ETA*ZET*I
     2CI(15,I)+ZET*ZET*ICI(17,I)+XSI*XSI*ZET*ICI(18,I)+2.0*XSI*ETA*ZET*I
     3CI(19,I)+XSI*ZET*ZET*ICI(20,I)
      DNZET(I)=ICI(4,I)+ETA*ICI(6,I)+XSI*ICI(7,I)+2.0*ZET*ICI(10,I)+XSI*
     1ETA*ICI(11,I)+XSI*XSI*ICI(13,I)+ETA*ETA*ICI(15,I)+2.0*XSI*ZET*ICI(
     216,I)+2.0*ETA*ZET*ICI(17,I)+XSI*ETA*XSI*ICI(18,I)+XSI*ETA*ETA*ICI(
     319,I)+2.0*XSI*ETA*ZET*ICI(20,I)
    3 CONTINUE
      DO 4 I=1,20
      AJ11=AJ11+DNXSI(I)*XX(I)
      AJ12=AJ12+DNXSI(I)*YY(I)
      AJ13=AJ13+DNXSI(I)*ZZ(I)
      AJ21=AJ21+DNETA(I)*XX(I)
      AJ22=AJ22+DNETA(I)*YY(I)
      AJ23=AJ23+DNETA(I)*ZZ(I)
      AJ31=AJ31+DNZET(I)*XX(I)
      AJ32=AJ32+DNZET(I)*YY(I)
      AJ33=AJ33+DNZET(I)*ZZ(I)
    4 CONTINUE
      DJ=AJ11*AJ22*AJ33+AJ12*AJ23*AJ31+AJ13*AJ21*AJ32-AJ12*AJ21*AJ33-AJ1
     11*AJ23*AJ32-AJ13*AJ22*AJ31
      DETJ(J)=DJ/512.0
      IF (ABS(DJ).LT.0.1E-16) GO TO 8
      RJ11=1.0/DJ*(AJ22*AJ33-AJ23*AJ32)
```

```
      RJ22=1.0/DJ*(AJ11*AJ33-AJ13*AJ31)
      RJ33=1.0/DJ*(AJ11*AJ22-AJ12*AJ21)
      RJ21=1.0/DJ*(-AJ21*AJ33+AJ23*AJ31)
      RJ31=1.0/DJ*(AJ21*AJ32-AJ22*AJ31)
      RJ32=1.0/DJ*(-AJ11*AJ32+AJ12*AJ31)
      RJ12=1.0/DJ*(-AJ12*AJ33+AJ13*AJ32)
      RJ13=1.0/DJ*(AJ12*AJ23-AJ13*AJ22)
      RJ23=1.0/DJ*(-AJ11*AJ23+AJ13*AJ21)
      DO 5 I=1,6
      DO 5 II=1,3
5     BB(I,II)=0.0
      DO 7 I=1,20
      DNX=RJ11*DNXSI(I)+RJ12*DNETA(I)+RJ13*DNZET(I)
      DNY=RJ21*DNXSI(I)+RJ22*DNETA(I)+RJ23*DNZET(I)
      DNZ=RJ31*DNXSI(I)+RJ32*DNETA(I)+RJ33*DNZET(I)
      BB(1,1)=DNX
      BB(2,2)=DNY
      BB(3,3)=DNZ
      BB(4,1)=DNY
      BB(4,2)=DNX
      BB(5,2)=DNZ
      BB(5,3)=DNY
      BB(6,1)=DNZ
      BB(6,3)=DNX
      JJ=3*I-2
      K=3*I-1
      L=3*I
      DO 6 M=1,6
      BA(M,JJ)=BB(M,1)
      BA(M,K)=BB(M,2)
      BA(M,L)=BB(M,3)
6     CONTINUE
```

```
7       CONTINUE
        GO TO 10
8       WRITE (6,9) J,INEL
9       FORMAT (//,5X,50HDETERMINANT OF JACOBIAN FOR INTEGRATION POINT NR
       1=,I5,14H  ELEMENT NR =,I5,9H  IS ZERO)
        GO TO 13
10      CONTINUE
        IF (J.LE.6) WF=BC
        IF (J.GT.6) WF=CC
        DO 12 I=1,60
        SS=0.0
        DO 11 K=1,6
        SS=SS+BA(K,I)*SIGIN(J,K)
11      CONTINUE
        FE(I)=FE(I)+SS*WF*DETJ(J)
12      CONTINUE
13      CONTINUE
        DO 14 I=1,20
        IK=IX(INEL,I)
        IF (CODE(IK).LE.7) GO TO 14
        NN=3*I-2
        FY=FE(NN+1)
        FZ=FE(NN+2)
        TANALF=1.732051/2.0
        ALFA=ATAN(TANALF)
        AR22=COS(ALFA)
        AR33=AR22
        AR23=-SIN(ALFA)
        AR32=-AR23
        FE(NN+1)=FY*AR22+FZ*AR23
        FE(NN+2)=FY*AR32+FZ*AR33
14      CONTINUE
```

```
      DO 23 I=1,20
      N=IX(INEL,I)
      NN=3*I-2
      COD=CODE(N)
      IF (CODE(N).GT.7) COD=CODE(N)-8
      NI=3*N-2
      NK=3*N-1
      NL=3*N
      IF (COD.EQ.0) GO TO 22
      GO TO (15,16,17,18,19,20,21), COD
15    G(NK)=G(NK)+FE(NN+1)
      G(NL)=G(NL)+FE(NN+2)
      GO TO 23
16    G(NL)=G(NL)+FE(NN+2)
      GO TO 23
17    G(NK)=G(NK)+FE(NN+1)
      GO TO 23
18    G(NI)=G(NI)+FE(NN)
      G(NL)=G(NL)+FE(NN+2)
      GO TO 23
19    G(NI)=G(NI)+FE(NN)
      G(NK)=G(NK)+FE(NN+1)
      GO TO 23
20    G(NI)=G(NI)+FE(NN)
21    GO TO 23
22    G(NI)=G(NI)+FE(NN)
      G(NK)=G(NK)+FE(NN+1)
      G(NL)=G(NL)+FE(NN+2)
23    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SIGEQU (SIGM,SIGEQ)
C
C     FIND EQUIVALENT OR UNIAXIAL STRESS FROM THE SIX COMPONETS OF SIGM.
C
      DIMENSION SIGM(6)
      A=SIGM(1)-SIGM(2)
      B=SIGM(2)-SIGM(3)
      C=SIGM(3)-SIGM(1)
      D=SIGM(4)
      E=SIGM(5)
      F=SIGM(6)
      SIGE=0.5*(A*A+B*B+C*C)+3.0*(D*D+E*E+F*F)
      SIGEQ=SQRT(SIGE)
      RETURN
      END
```

```
      SUBROUTINE STRSTR (INEL,EPSUP,YS,A,ICOLL)
C
C     SUBROUTINE STRSTR RELATES UNIAXIAL PLASTIC STRAIN TO STRESS.
C     A IS THE SLOPE OF THE CURVE. A CHECK IS MADE WHETHER COLLAPSE
C     STRAIN IS REACHED.
C
      INTEGER CODE
      COMMON /DAT/ X(1000),Y(1000),Z(1000),CIP(14,3),ICI(20,20),CODE(100
     10),IX(200,22),KODE(1000),NUMNP,NUMEL
      COMMON /LAS/ E(8,8),D(6,6)
      MTYPE=IX(INEL,21)
      EP=EPSUP
      YS=E(5,MTYPE)
      IF (EP.GT..06) ICOLL=1
      A=0.0
      RETURN
      END
```

```
      SUBROUTINE TITLE
      COMMON /HAD/ HED(18),IPAGE,LINE
      WRITE (6,1) IPAGE
1     FORMAT (1H1,3X,39H60-DOF GENERAL HEXAHEDRON ELEMENT, HATT,50X,4HPA
     1GE,I3)
      WRITE (6,2) HED
2     FORMAT (1H0,10X,18A4)
      IPAGE=IPAGE+1
      LINE=0
      RETURN
      END
```

```
      SUBROUTINE RESOL (A,B,G,MAXB,NTDOF,NEQB,MB,LL,NBLOCK,NSB,NORG,NRST
     1)
C
C     MODIFIED SOLUTION ROUTINE FROM USOL. THE TRIANGULIZED EQUATIONS
C     RESIDE ON UNIT 2. LOAD VECTOR G HAS TO BE MODIFIED FIRST.
C     THEN THE BACKSUBSTITUTION CAN BE PERFORMED.
C
      IMPLICIT REAL * 8 (A-G,O-Z)
      DIMENSION A(NSB), B(NSB), G(NTDOF), MAXB(NEQB)
      NBR=(MB-1)/NEQB+1
      NMB=NEQB*MB
      DO 5 N=1,NBLOCK
      READ (NORG) A,MAXB
      DO 2 I=1,NEQB
      JJ=NEQB+I
      JK=MAXB(I)
      C=A(I)
      DO 1 J=JJ,JK,NEQB
      A(J)=A(J)*C
1     CONTINUE
2     CONTINUE
      DO 4 I=1,NEQB
      NN=(N-1)*NEQB+I
      C=A(I)
      G(NN)=G(NN)/C
      IF (NN.EQ.NTDOF) GO TO 5
      NK=NN+MB
      JJ=0
      NS=NN+1
      DO 3 J=NS,NK
      KK=I+NEQB+JJ*NEQB
      IF (KK.GT.MAXB(I)) GO TO 4
```

```
      G(J)=G(J)-G(NN)*A(KK)
      JJ=JJ+1
3     CONTINUE
4     CONTINUE
5     CONTINUE
      LS=LL*NEQB
      NEB=NEQB*(NBR+1)
      NUM=NBR*NEQB
      MAX=NEB*LL
      DO 6 I=1,MAX
6     B(I)=0.D0
      REWIND NRST
      DO 14 N=1,NBLOCK
      MI=NORG
      BACKSPACE MI
      READ (MI) A,MAXB
      DO 7 II=1,NEQB
      IL=NSB-NEQB+II
      IM=(NBLOCK-N)*NEQB+II
      IF (IM.GT.NTDOF) GO TO 7
      A(IL)=G(IM)
7     CONTINUE
      BACKSPACE MI
      DO 8 L=1,LL
      K=L*NEB
      DO 8 J=1,NUM
      I=K-NEQB
      B(K)=B(I)
8     K=K-1
      I=NMB
      DO 9 L=1,LL
      K=(L-1)*NEB
```

```
      DO 9 J=1,NEQB
      I=I+1
      K=K+1
      B(K)=A(I)
 9    CONTINUE
      DO 12 I=1,NEQB
      J=NEQB+1-I
      MAX=MAXB(J)
      IF (A(J).EQ.0.DO) GO TO 12
      DO 11 L=1,LL
      KK=J+(L-1)*NEB
      JJ=KK+1
      IL=J+NEQB
      C=B(KK)
      DO 10 II=IL,MAX,NEQB
      C=C-A(II)*B(JJ)
 10   JJ=JJ+1
 11   B(KK)=C
 12   CONTINUE
      I=0
      DO 13 L=1,LL
      K=(L-1)*NEB
      DO 13 J=1,NEQB
      K=K+1
      I=I+1
 13   A(I)=B(K)
      WRITE (NRST) (A(I),I=1,LS)
 14   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ELAS (MTYPE)
C
C     CALCULATION OF ELASTICITY MATRIX (D)
C
      COMMON /LAS/ E(8,8),D(6,6)
      EE=E(1,MTYPE)
      XNU=E(2,MTYPE)
      DO 1 I=1,6
      DO 1 J=1,6
1     D(I,J)=0.0
      CONST=EE*(1.0-XNU)/((1.0+XNU)*(1.0-2.0*XNU))
      D(1,1)=CONST
      D(1,2)=CONST*XNU/(1.0-XNU)
      D(1,3)=D(1,2)
      D(2,2)=CONST
      D(2,3)=D(1,2)
      D(3,3)=CONST
      D(4,4)=CONST*(1.0-2.0*XNU)/(2.0*(1.0-XNU))
      D(5,5)=D(4,4)
      D(6,6)=D(4,4)
      D(2,1)=D(1,2)
      D(3,1)=D(1,3)
      D(3,2)=D(2,3)
      RETURN
      END
```

```
      SUBROUTINE PLASTR (INEL,DSIG,SIGIN,ICOLL)
C
C     THE TOTAL STRESS INCREMENT AND THE CORRESPONDING INITIAL
C     STRESSES ARE FOUND. TOTAL STRESSES AND PLASTIC STRAINS
C     ARE UPDATED.
C
      INTEGER CODE
      COMMON /DAT/ X(1000),Y(1000),Z(1000),CIP(14,3),ICI(20,20),CODE(100
     10),IX(200,22),KODE(1000),NUMNP,NUMEL
      COMMON /STR/ SIG(350,6),EUP(350)
      COMMON /LAS/ E(8,8),D(6,6)
      DIMENSION DSIG(14,6), SIGIN(14,6), SIGM(6), ALC(6), DLC(6), DELSIG
     1(6)
      MTYPE=IX(INEL,21)
      CALL ELAS (MTYPE)
      DO 1 I=1,14
      DO 1 J=1,6
1     SIGIN(I,J)=0.0
      DO 19 I=1,14
      N=(INEL-1)*14+I
C
C     DELSIG IS ELASTIC STRESS INCREMENT FOR THE PARTICULAR INTEGRATION
C
      DO 2 J=1,6
      DELSIG(J)=DSIG(I,J)
2     CONTINUE
C
C     CHECK WHETHER INTEGRATION POINT YIELDED PREVIOUSLY.
C
      IF (EUP(N).GT.0.1E-11) GO TO 7
      DO 3 J=1,6
      SIGM(J)=SIG(N,J)+DSIG(I,J)
```

```
3     CONTINUE
C
C     CHECK YIELD CONDITION FOR SIG+EL.INCR. OF SIG.
C
      CALL SIGEQU (SIGM,SIGEQ)
      EPSUP=EUP(N)
      CALL STRSTR (INEL,EPSUP,YS,A,ICOLL)
      F1=SIGEQ-YS
C
C     IF NO YIELD ENCOUNTERED TRANSFER TO END OF LOOP WHERE STRESS IS UP
C
      IF (F1.LE.0.0) GO TO 16
      DO 4 J=1,6
      SIGM(J)=SIG(N,J)
4     CONTINUE
C
C     CHECK YIELD CONDITION FOR SIG
C
      CALL SIGEQU (SIGM,SIGEQ)
      F0=SIGEQ-YS
      WRITE (6,5) INEL,I,F0,F1,YS
5     FORMAT (5X,7HEL.NO.=,I5,10HINT.POINT=,I5,3HF0=,E15.6,3HF1=,E15.6,3
     1HYS=,E15.6)
      R=-F0/(F1-F0)
C
C     IF INTEGRATION POINT YIELDS DURING THE INCREMENT, UPDATE SIG TO YI
C     LEVEL.
C     DELSIG IS (1-R)*ELASTIC STRESS INCREMENT.
C
      DO 6 J=1,6
      SIG(N,J)=SIG(N,J)+R*DSIG(I,J)
      DELSIG(J)=(1.0-R)*DSIG(I,J)
```

```
6     CONTINUE
7     CONTINUE
      EPSUP=EUP(N)
      CALL STRSTR (INEL,EPSUP,YS,A,ICOLL)
C
C     FORMULATE LOWER CASE A-VECTOR.
C
      SM=(SIG(N,1)+SIG(N,2)+SIG(N,3))/3.0
      SX=SIG(N,1)-SM
      SY=SIG(N,2)-SM
      SZ=SIG(N,3)-SM
      SXY=SIG(N,4)
      SYZ=SIG(N,5)
      SZX=SIG(N,6)
      ALC(1)=1.5*SX/YS
      ALC(2)=1.5*SY/YS
      ALC(3)=1.5*SZ/YS
      ALC(4)=3.0*SXY/YS
      ALC(5)=3.0*SYZ/YS
      ALC(6)=3.0*SZX/YS
C
C     FORMULATE LOWER CASE D-VECTOR.
C
      DO 9 J=1,6
      DLC(J)=0.0
      DO 8 K=1,6
      DLC(J)=DLC(J)+D(J,K)*ALC(K)
8     CONTINUE
9     CONTINUE
      BET=0.0
C
C     FIND BETA.
```

```
C
      DO 10 J=1,6
      BET=BET+ALC(J)*DLC(J)
10    CONTINUE
      DLAM=0.0
C
C     FIND DLAMBDA.
C
      DO 11 J=1,6
      DLAM=DLAM+ALC(J)*DELSIG(J)
11    CONTINUE
      WRITE (6,12) DLAM,A,BET
12    FORMAT (5X,5HDLAM=,E15.6,4H  A=,E15.6,6H  BET=,E15.6)
      DLAM=DLAM/(A+BET)
C
C     IF DLAMBDA NEGATIVE, UNLOADING OCCURS, ONLY ELASTIC CHANGES.
C     ONLY PERMISSIBLE IF EUP(N) WAS GREATER THAN ZERO.
C
      IF (DLAM.LE.0.0) GO TO 16
C
C     FORMULATE INITIAL STRESS VECTOR.
C
      DO 13 J=1,6
      SIGIN(I,J)=DLAM*DLC(J)
13    CONTINUE
      DO 14 J=1,6
      DELSIG(J)=DELSIG(J)-SIGIN(I,J)
14    CONTINUE
C
C     UPDATE TOTAL STRESS.
C
      DO 15 J=1,6
```

```
      SIG(N,J)=SIG(N,J)+DELSIG(J)
   15 CONTINUE
C
C     UPDATE UNIAXIAL PLASTIC STRAIN.
C
      EUP(N)=EUP(N)+DLAM
      GO TO 18
   16 CONTINUE
C
C     UPDATE FOR ELASTIC ONLY STRESS INCREMENT OR FOR UNLOADING.
C
      DO 17 J=1,6
      SIG(N,J)=SIG(N,J)+DSIG(I,J)
   17 CONTINUE
   18 CONTINUE
   19 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PRINTS (NUMEL)
C
C     PRINT UPDATED STRESSES AND STRAINS
C
      COMMON /STR/ SIG(350,6),EUP(350)
      DIMENSION SIGM(6)
      CALL TITLE
      DO 6 I=1,NUMEL
      WRITE (6,1) I
1     FORMAT (/,10X,48HSTRESSES AT ALL INTEGRATION POINTS OF ELEMENT NO,
     1I5,/)
      WRITE (6,2)
2     FORMAT (/,5X,10HINT. POINT,5X,7HSIG(XX),7X,7HSIG(YY),7X,7HSIG(ZZ),
     17X,7HSIG(XY),7X,7HSIG(YZ),7X,7HSIG(ZX),8X,5HSIGEQ,8X,3HEUP,//)
      DO 5 J=1,14
      N=(I-1)*14+J
      DO 3 K=1,6
3     SIGM(K)=SIG(N,K)
      CALL SIGEQU (SIGM,SIGEQ)
      WRITE (6,4) J,(SIG(N,K),K=1,6),SIGEQ,EUP(N)
4     FORMAT (5X,I5,4X,8E14.3)
5     CONTINUE
6     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ACCEL (DIST,XY,YZ,ZX,AVG,NUMIT)
      INTEGER COD,CODE
      COMMON /DAT/ X(1000),Y(1000),Z(1000),CIP(14,3),ICI(20,20),CODE(100
     10),IX(200,22),KODE(1000),NUMNP,NUMEL
      COMMON /ACC/ AC(3000)
      DIMENSION DIST(3000), XY(1000), YZ(1000), ZX(1000)
      DO 11 I=1,NUMNP
      II=3*I-2
      IJ=3*I-1
      IK=3*I
      AC(II)=1.0
      AC(IJ)=1.0
      AC(IK)=1.0
      COD=CODE(I)
      IF (CODE(I).GT.7) COD=CODE(I)-8
      IF (KODE(I).GT.0) COD=KODE(I)
      IF (COD.EQ.0) GO TO 7
      GO TO (1,2,3,4,5,6,8), COD
1     AC(IJ)=1.0+YZ(I)/DIST(IJ)
      AC(IK)=1.0+ZX(I)/DIST(IK)
      GO TO 9
2     AC(IK)=1.0+ZX(I)/DIST(IK)
      GO TO 9
3     AC(IJ)=1.0+YZ(I)/DIST(IJ)
      GO TO 9
4     AC(II)=1.0+XY(I)/DIST(II)
      AC(IK)=1.0+ZX(I)/DIST(IK)
      GO TO 9
5     AC(II)=1.0+XY(I)/DIST(II)
      AC(IJ)=1.0+YZ(I)/DIST(IJ)
      GO TO 9
6     AC(II)=1.0+XY(I)/DIST(II)
```

```
      GO TO 9
7     AC(II)=1.0+XY(I)/DIST(II)
      AC(IJ)=1.0+YZ(I)/DIST(IJ)
      AC(IK)=1.0+ZX(I)/DIST(IK)
8     CONTINUE
9     CONTINUE
      DIST(II)=XY(I)
      DIST(IJ)=YZ(I)
      DIST(IK)=ZX(I)
      IF (NUMIT.GT.1) GO TO 10
      AC(II)=1.0
      AC(IJ)=1.0
      AC(IK)=1.0
10    CONTINUE
11    CONTINUE
      IND=0
      ASUM=0.0
      NTDOF=3*NUMNP
      DO 14 I=1,NTDOF
      IF (AC(I).GT.1.0) GO TO 12
      GO TO 13
12    ASUM=ASUM+AC(I)
      IND=IND+1
13    CONTINUE
14    CONTINUE
      IF (ASUM.LT.1.0) GO TO 17
      AVG=ASUM/IND
      IF (AVG.GT.2.0) GO TO 15
      GO TO 18
15    WRITE (6,16)
16    FORMAT (5X,59HAVERAGE ACCELERATION FACTOR GREATER THAN 2.0, SET TO
     1    1.0)
```

```
17    AVG=1.0
18    CONTINUE
      WRITE (6,19) AVG,IND
19    FORMAT (//,5X,30HAVERAGE ACCELERATION FACTOR IS,E15.5,I5,/)
      DO 20 I=1,NUMNP
      XY(I)=XY(I)*AVG
      YZ(I)=YZ(I)*AVG
      ZX(I)=ZX(I)*AVG
20    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LOAD (GD,AVG,NTDOF)
      REAL * 8 GD
      COMMON /ACC/ AC(3000)
      COMMON /GEE/ G(3000)
      COMMON /GEO/ GE(3000)
      DIMENSION GD(3000)
      DO 1 I=1,NTDOF
      GD(I)=G(I)-GE(I)*(AVG-1.0)
      GE(I)=GD(I)
    1 CONTINUE
      RETURN
      END
```

The vita has been removed from the scanned document

THREE-DIMENSIONAL FINITE-ELEMENT ANALYSIS
INCLUDING ELASTIC-PLASTIC MATERIAL RESPONSE

by

Fritz Hatt

(ABSTRACT)

The development of a linear finite-element program using an isoparametric 60-DOF hexahedron is presented. As a special feature dictated by programming considerations, the enforcement of boundary conditions and the modification of the stiffness matrix due to a rotated coordinate system for selected nodal points is carried out on the element level. Numerical integration is performed with the aid of a 14-point integration rule.

The nonlinear program is capable of modelling elastic-plastic material behavior. It is based on the constant stiffness iteration employing the initial stress concept.

The iterative procedure is accelerated by a new scheme termed uniform acceleration.

Three practical applications are presented. The thick-walled cylinder under internal pressure and the transversely loaded thin plate on simple supports are analyzed under the assumption of ideal plasticity. Comparisons with existing solutions are also made. For the third example, an aluminum-epoxy fiber composite is subjected to a uniform temperature difference. Both materials are characterized by strain

hardening. The effect of a free surface perpendicular to the fiber direction is shown.